8514/A GRAPHICS FOR THE PS/2 ▶

POP-UPS UNDER OS/2

MAY 1988

VOL. 6 NO. 5 $3.95

PC TECH JOURNAL®

FOR SYSTEMS DEVELOPERS AND INTEGRATORS

05

0 71486 02166 7

Buy Our Tools, And We'

Introducing Emerald Bay. The breakthrough database server technology for developing single and multi-user applications. Emerald Bay provides your programs a common data storage and retrieval method which allows data to be transparently shared across multiple and diverse applications.

And when you buy one of our tools for "C," dBASE™ or Lotus® developers, we'll give you the personal engine—free. No royalties to pay, no licenses to sign.

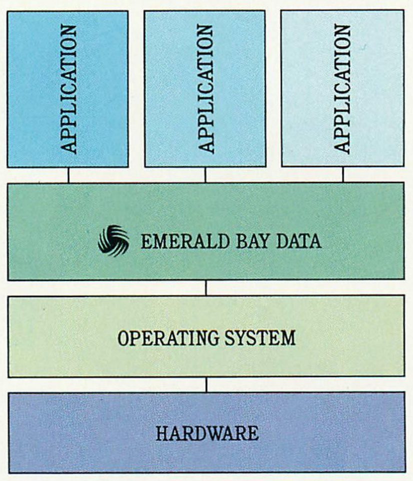
Developed by Wayne Ratliff, the creator of dBASE, Emerald Bay is much more than just another DBMS product, it's an entirely new way to manage data. It's designed to provide an open platform for developing applications in several languages and environments, while Emerald Bay maintains data security, concurrency and integrity.

How The Engine Works

Before, data couldn't be readily shared between applications. But with Emerald Bay, PC applications each share a common data storage and retrieval method. And although the functions of the applications may vary widely, any one application can share another's data transparently; there is no data conversion or translation necessary.

When a PC is an intelligent workstation on a LAN, the Emerald Bay database server technology controls all data

Emerald Bay Architecture



security and integrity, including transaction logging with rollback. An application simply makes a request, which is sent to the engine. There, only the essential data is sent back to the workstation. The result is vastly reduced network traffic and faster data access times.

How You Work With The Tools

With the tools we provide, you can easily develop Emerald Bay applications immediately in your familiar development environment.

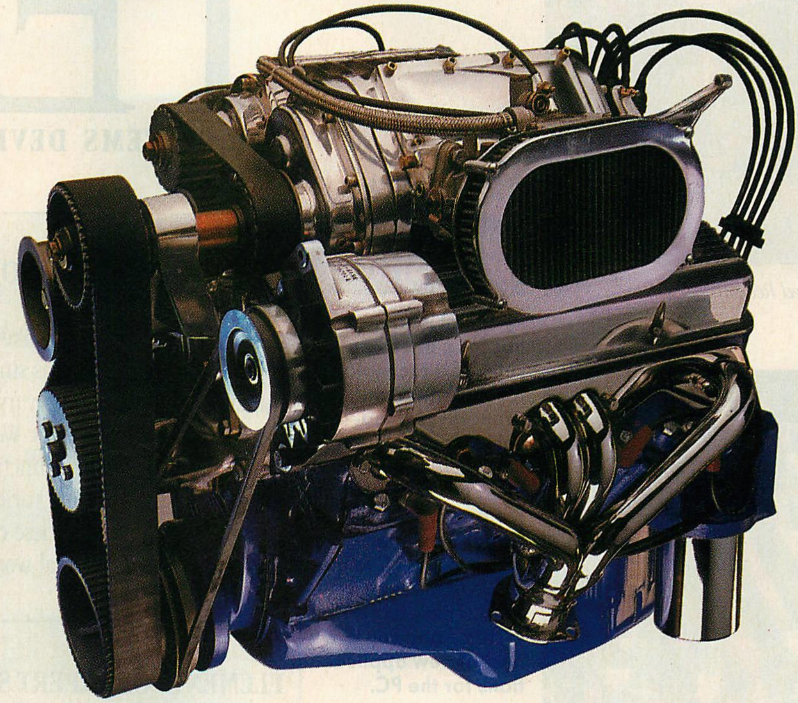
Emerald Bay technology handles the usually code-intensive management of data, so you can concentrate on what you do best—developing applications.

The ***Developers Toolkit for "C"*** includes well-documented, easy to use "C" libraries that give you the power to create advanced applications, without the effort usually associated with designing and coding a database "backend."

Eagle is Emerald Bay's sophisticated dBASE-like programming language. As the logical evolution of database language, Eagle introduces advanced features, routines and language components, including a compiler, network commands, user-defined functions in "C" and Assembly and automatic index maintenance.

Summit is an "add-in" database management system for Lotus 1-2-3, which gives you sophisticated data manipulation and analysis commands. All three of Emerald Bay's development tools come with the Core Components which include Report Writer, Forms Generator,

ll Give You The Engine.



an Import/Export facility and the Database Administrator.

The ***Emerald Bay Database Server*** is the heart of the multi-user Emerald Bay technology. Its client/server architecture is superior to current implementations of LAN/DBMS products, and increases total system throughput, while reducing network traffic and access times.

Finally, while providing a path to other operating systems such as OS/2, Macintosh and UNIX, Emerald Bay is a microcomputer-based technology that optimizes your *current* hardware investment.

Free Technical Seminars

We're hosting a series of free Emerald Bay Technical Seminars during April and May in cities across the country. It's your chance to see Wayne Ratliff demonstrate the capabilities of Emerald Bay in person, as well as get some practical experience with the technology yourself.

Call us toll-free at 1-800-777-2027 (and ask for Sandra) for the date and location of the seminar nearest you. Space is limited, so be sure to reserve your seat today.

Emerald Bay. Advanced database server technology. Available *now.*

Emerald Bay Engine Specifications

Data Storage

· Max. databases	No limit
· Max. tables per database	1000
· Max. fields per table	800
· Max. field width	512 characters
· Max. records per table	No limit
· Max. width of records	10,000 bytes (no limit on ext. fields)
· Max. open databases	7 (MS-DOS limitation)

Index Storage

- Composite keys supported
- Mixed data type keys allowed
- Keys of up to 100 bytes in length
- Automatic index maintenance
- Ascending and descending keys
- Case independent keys
- Automatic table indexing on record number

Security And Integrity Features

- Access permissions by Read, Write, Delete, Add and Grant
- All five access permissions work on tables and objects
- Read, Write and Grant access permissions operate at field level
- All data other than binary fields can be encrypted
- Transaction logging, with commit and rollback functions
- Full security functions at field and table level
- Optional data encryption at field level

System Requirements

- MS-DOS 3.1 or greater
- Network database server or Single-user computer: PC XT, AT, PS/2 or 386 compatible, 640K, Hard Disk
- Workstation on LAN: PC, XT, AT, PS/2 or 386 compatible, 640K
- NetBIOS compatible networks supported

EMERALD BAY

MIGENT™

865 Tahoe Blvd., Call Box 6, Incline Village, NV 89450

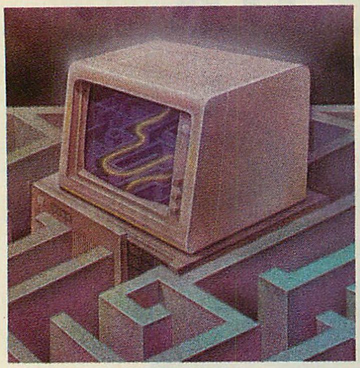
Trademarks pending: Emerald Bay, Eagle, Summit (Migent, Inc.)

CIRCLE NO. 161 ON READER SERVICE CARD

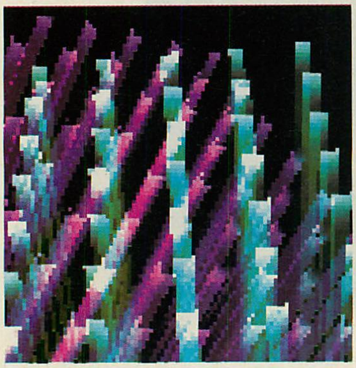
MAY 1988

PC TECH JOURNAL

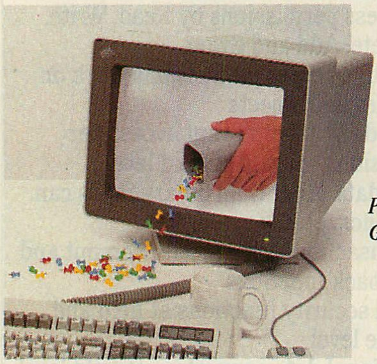
FOR SYSTEMS DEVELOPERS AND INTEGRATORS



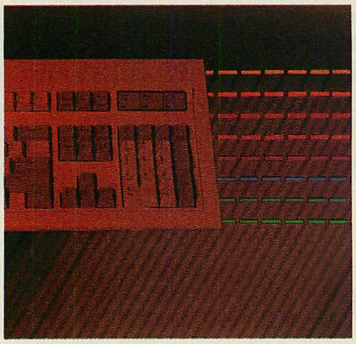
Computerized Reasoning **44**



db_VISTA'S Network Approach **134**



Pixels with Panache **102**



OS/2's Answer to TSRs **84**

COVER SUITE: EXPERT SYSTEMS

Artificial intelligence has long been considered a technology for the future and, primarily, for machines larger than the PC. It is, however, pushing into the present as AI techniques filter down to the PC in the guise of expert systems. These expert systems open up a host of new applications for the PC.

OPERATING ENVIRONMENTS

SYSTEMS FORUM UPDATE

TECH PC JOURNAL Systems Forum '88 SAN FRANCISCO

Plans for the 1988 *PC Tech Journal* Systems Forum, June 13–15 at the St. Francis Hotel in San Francisco, are in the final stages.

The keynote address, "Computing into the 1990s (and how to get there from here)" will be given Monday by Dr. James Nestor of Ernst & Whinney. Steve Ballmer of Microsoft will give the Tuesday luncheon speech, "Why the Graphical User Interface?"

The lineup of panelists includes Teri Myers (Quarterdeck), Safi Qureshey (AST), Gary Stimac (Compaq), Danielle Barr (Bank of Boston), Umang Gupta (Gupta Technologies), Bob Epstein (Sybase), Peter Coffee (Aerospace Corp.), Scott Tucker (Lotus), Tyrone Pike (LAN Systems), and Nat Goldhaber (TOPS).

To register for Systems Forum '88, or for more information, call 800/644-PCTJ or write to *PC Tech Journal* Systems Forum, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. —*WF*

DEPARTMENTS

Cover illustration · Andy Levine

TECH PC JOURNAL

VOL. 6, NO. 5

PUBLISHER: *Donald J. Byrnes*
EDITORIAL DIRECTOR: *Will Fastie*
EDITOR: *Julie Anderson*

EDITORIAL
EXECUTIVE EDITORS: *Susan Holly, David Methvin*
MANAGING EDITOR: *Gail Shaffer*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Maxine Fontana, Philip N. Hisley, Steven S. King, Ted Mirecki*
ASSOCIATE EDITORS: *Douglas Tallman, Jordene Zeimetz*
SENIOR COPY EDITOR: *Bruce Ansley*
COPY EDITOR: *Judith Estrin*
EDITORIAL PRODUCTION ASSISTANT: *Katherine M. Kiwak*
NEW PRODUCTS EDITOR: *Carole Eyring*
LAB TECHNICIAN: *Todd Bannar*
EDITORIAL ASSISTANT: *Barbara Smith*
BUSINESS MANAGER: *Trish Ledbetter*
EDITORIAL SECRETARY: *Beth Sauberlich*
RECEPTIONIST: *Tobie J. Ketterer*
CONTRIBUTING EDITORS: *Steven Armbrust, Dave Browning, David Claiborne, Peter C. Coffee, Richard M. Foard, Ted Forgeron, Thomas V. Hoffmann, Max Stul Oppenheimer*

ART & PRODUCTION
ART DIRECTOR: *Sharon Reuter*
ASSOCIATE ART DIRECTOR: *Courtney Barone*
ASSISTANT ART DIRECTOR: *Crystal Hopkins*
ART ASSISTANT: *Cheri Glover*
PRODUCTION MANAGER: *Alison Regan Mrohs*
CONTRIBUTING ARTISTS: *Maciek Albrecht, David Povilaitis*

ADVERTISING SALES
ADVERTISING DIRECTOR/EAST: *Rita Burke*
ADVERTISING DIRECTOR/WEST: *Phyllis Egan*
NATIONAL TELEMARKETING MANAGER: *Arlene Braithwaite*
MARKETING DIRECTOR: *Marti Cunha*
ASSISTANT TO THE PUBLISHER: *Lucy Frazer*
ADVERTISING COORDINATOR: *JeanMarie Donlin*
MARKETING COORDINATOR: *Kimberly Schroeder*
MARKETING ASSISTANT: *Debbie Lintner*
SALES SECRETARY: *Mary Barrow*
DISTRICT MANAGERS: *Bill Barney—Southeast and Eastern Canada; Melinda Maginn—New England; John Blake—Mid-Atlantic; Mike Leonard—Midwest; Steve Moorman, Gayl Sorota, Deborah Gisonni, Elaine Ebner—West Coast*
ACCOUNT REPRESENTATIVES: *Mary Martin—Southeast and New England; Nannette Vilushis—Midwest and Mid-Atlantic; Patti Abiswede, Donna Wood Tyrus—West Coast; Classified Advertising Director—Kathryn Cumberlander*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*
RETAIL SALES MANAGER: *Carol Benedetto*

GROUP VICE PRESIDENT: *J. Scott Briggs*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Co.*
PRESIDENT: *Kenneth H. Koppel*
GROUP VICE PRESIDENTS: *J. Scott Briggs, William Lohse*
SENIOR VICE PRESIDENT, Marketing: *Paul H. Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Research: *Marian O. White*
VICE PRESIDENT, Classified Advertising: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation: *Bert Lacy*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
DIRECTOR OF PLANNING: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
CHAIRMAN: *Philip B. Korsant;* PRESIDENT: *Kenneth H. Koppel;* SENIOR VICE PRESIDENTS: *Philip Sine, Hugh Tietjen;* VICE PRESIDENTS: *William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Laurence Usdin;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC Tech Journal, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. FAX (group 3): 301/740-8809. MCIMail: PCTECH. PCTECHline: 301/740-8383. Telex: 6502565932 MCI.

ADVERTISING OFFICES
Corporate/Southeast/Canada: Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044; 301/740-8300. New England: 144 Watchung Avenue, Upper Montclair, NJ 07043; 201/509-8148. Mid-Atlantic: 89 Pearsall Place, Merrick, NY 11566; 516/546-7505. Midwest: Suite 1400, 180 N. Michigan Avenue, Chicago, IL 60601; 312/346-2600. West Coast: 3460 Wilshire Blvd., Los Angeles, CA 90010; 213/387-2100. 110 Marsh Drive, Foster City, CA 94404; 415/378-5580.

SUBSCRIPTION INQUIRIES
PC Tech Journal, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing. One Park Avenue, 4th floor, New York, NY 10016.

PC Tech Journal (ISSN 0738-0194) is published by Ziff-Davis Publishing Co., a division of Ziff Communications Co., One Park Ave., New York, NY 10016. Published monthly except semi-monthly in December. Subscription rate is $34.97 for one year (13 issues). Additional postage for Canada and Foreign is $.60/copy or $8.00/year. Second-class postage paid at New York, NY, and at additional mailing offices. POSTMASTER: Send address changes to PC Tech Journal, P.O. Box 2968, Boulder, CO 80321.

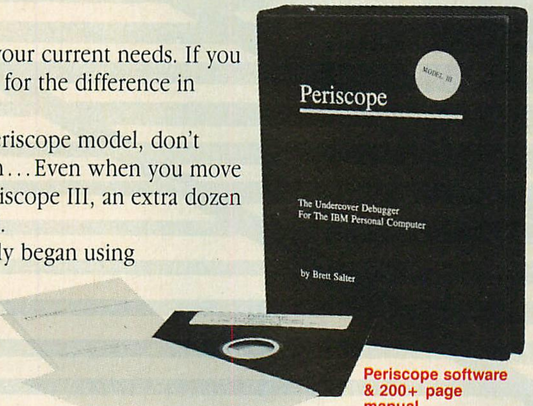
PERISCOPE™ POWER

...Keeps you going full steam ahead when other debuggers let you down! With four models to pick from, you'll find a Periscope that has just the power you need.

Start with the model that fits your current needs. If you need more horsepower, upgrade for the difference in price plus $10!

When you move to another Periscope model, don't worry about having a lot to learn... Even when you move to the most powerful model, Periscope III, an extra dozen commands are all that's involved.

A Periscope I user who recently began using Periscope III writes, *"I like the fact that within the first half hour of use I was debugging my program instead of learning to use the debugger."*

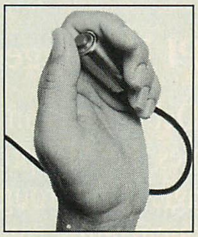


Periscope software & 200+ page manual

- **Periscope's software is solid, comprehensive, and flexible.** It helps you debug just about any kind of program you can write... thoroughly and efficiently.

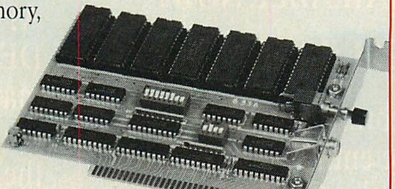
Periscope's the answer for debugging device-drivers, memory-resident, non-DOS, and interrupt-driven programs. Periscope works with any language, and provides source and/or symbol support for programs written in high-level languages and assembler.

- **Periscope's hardware adds the power to solve the really tough debugging problems.** The break-out switch lets you break into the system any time. You can track down a bug instantly, or just check what's going on, without having to reboot or power down and back up. That's really useful when your system hangs! The switch is included with Periscope I, Periscope II, and Periscope III.



Periscope Break-Out Switch

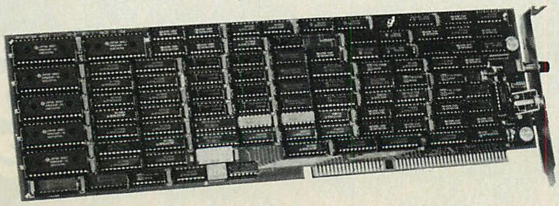
Periscope I has a board with 56K of write-protected RAM. The Periscope software resides in this memory, safe from run-away programs. DOS memory, where debugger software would normally reside, is thus freed up for your program.



Periscope I Board

Periscope III has a board with 64K of write-protected RAM, which performs the same function as the Periscope I protected memory. AND...

The Periscope III board adds another powerful dimension to your debugging. Its hardware breakpoints and real-time trace buffer let you track down bugs that a software-oriented debugger would take too long to find, or can't find at all!



Periscope III Board

What Periscope Users Like Best:

"I like the clean, solid design and the crash recovery."
Periscope I user

"I like the ability to break out of (a) locked up system!"
Periscope II user

"I am very impressed with Periscope II-X ... it has become my 'heavy duty' debugger of choice, especially if I need to work on a memory resident utility or a device driver."
Periscope II-X user

"... Periscope III is the perfect answer to the debugging needs of anyone involved in real-time programming for the PC ... The real time trace feature has saved me many hours of heartache already."
Periscope III user

- **Periscope I** includes a half-length board with 56K of write-protected RAM; break-out switch; software and manual for $345.
- **Periscope II** includes break-out switch; software and manual for $175.
- **Periscope II-X** includes software and manual (no hardware) for $145.
- **Periscope III** includes a full-length board with 64K of write-protected RAM, hardware breakpoints and real-time trace buffer; break-out switch; software and manual. Periscope III for machines running up to 8 MHz is $995; for machines running up to 10 MHz, $1095.

REQUIREMENTS: IBM PC, XT, AT, or close compatible (Periscope III requires hardware as well as software compatibility); DOS 2.0 or later; 64K available memory; one disk drive; an 80-column monitor.

Call us with your questions. We'll be happy to send you free information or help you decide on the model that best fits your needs.

Order Your Periscope, Toll-Free, Today!
800-722-7006
MAJOR CREDIT CARDS ACCEPTED

The
PERISCOPE
Company, Inc.

1197 PEACHTREE ST.
PLAZA LEVEL
ATLANTA, GA 30361
404/875-8080

CIRCLE NO. 215 ON READER SERVICE CARD

Finally, there's an SQL that gets back to BASIC. And COBOL. And C. And Pascal.

As a programmer, you've probably already faced it—the database dilemma. Do you use an SQL for easy database handling, or a true programming language for maximum power and flexibility?

Now you can do both with XQL,® the relational data management system from the developers of Btrieve.®

The Programmer's SQL. With XQL, you can access your data with the ease of Structured Query Language through simple subroutine calls from traditional programming languages. XQL supports standard SQL syntax, including subqueries, unions and security groups.

XQL Relational Primitive Operations. In addition, XQL lets you bypass the SQL level and perform highly efficient, relational primitive operations directly. You get all the functionality of a relational database model without the constraints of a 4th generation language.

Building on Btrieve. The heart of Novell's family of data management tools is Btrieve. By letting you access multiple records at a time, XQL adds a powerful dimension to Btrieve. XQL incorporates sophisticated data manipulation features which allow you to access data by field name, move forward or backwards through the database, compute fields from other fields or constants, and even work with composite records built from multiple, joined Btrieve files.

Like Btrieve, XQL offers features like multi-user support, fault tolerance, comprehensive documentation, and expert technical support. And you never pay royalties on your XQL applications.

Solve the database dilemma with XQL, the SQL that speaks your language. Only $795.* See your Authorized Novell Gold Reseller, or call us at (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVXQL3.

NOVELL

For software solutions, you should be seeing red.

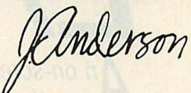
*Suggested retail price (US dollars) ©1988 Novell Inc., World Headquarters, 122 East 1700 South, Provo, Utah 84601 (801) 379-5900
Requires Btrieve 4.x and PC-DOS or MS-DOS 2.x, 3.x.

CIRCLE NO. 201 ON READER SERVICE CARD

JULIE ANDERSON

AI Escapes the Lab

Expert system shells open the door to using artificial intelligence techniques for commercial applications.



With expert system shells leading the way, artificial intelligence is working itself out of the science lab and into the real world. Using AI techniques, these shells simplify expert-systems development by providing a structure for organizing and using the collective experience in a given field. These expert systems can aid a physician in making complex diagnoses, put the knowledge of an engineer into the hands of a repairman, or extend the reach of a bank officer by authorizing loans in a branch office.

While expert system shells are an emerging technology, they share a common characteristic with more established data management systems: they provide a framework in which to define, manipulate, and report data. The difference is that most data management systems store data in relational tables, whereas expert system shells represent their data—the knowledge base—using rules, frames, or semantic nets. An expert system's inference engine gathers the appropriate assertions set forth in the knowledge base and uses predefined control strategies to manipulate the data, until it reaches a set of solutions—the query output. Expert systems can take data management one step further by manipulating data in a database, using guidelines from the knowledge base.

Unlike data managers, expert systems can deal with uncertain or incomplete data; thus, their output may be a set of answers with varying levels of confidence. For those who are suspicious of what appears to be a reasoning machine, be consoled that you can audit an expert system's conclusions. Most expert systems have an explanation feature that traces the steps that the computer follows to reach its conclusion.

Although, like data managers, expert systems could be written from scratch using traditional languages, the framework the shell provides dramatically shortens development time. In some cases, an expert system shell serves as a rapid prototyping tool. Once the system is proven, it can be rewritten in a traditional AI language such as LISP, or even in C.

Using a shell does not mean the expert, working alone, can build a successful and efficient expert system. As in data management, the end user who builds a database may end up with redundant data and too many or too few indexes. Likewise, it takes the hands of a professional who understands knowledge engineering and AI techniques to make an expert system come alive.

Building an expert system, therefore, is usually a team effort. The expert, or group of experts, joins forces with a knowledge engineer/developer to construct the knowledge base and define the control strategies that will best manipulate it. Besides determining how the knowledge is to be represented, a knowledge engineer must also extract the knowledge from the expert. Loading a knowledge base is more difficult than loading a database because knowledge is less tangible than data. A customer's name, address, and account number are more easily determined than the thought processes an account manager goes through when deciding whether or not to extend a customer's credit.

This month's cover suite focuses on the issues involved in developing expert systems using a shell. Because this may be an unfamiliar topic for many of our readers, author Tom Arcidiacono introduces expert system development terms and techniques in the first article "Computerized Reasoning" (p. 44). A consultant in knowledge-based systems development, Arcidiacono teaches computer science at the New York Institute of Technology.

Also in this cover suite, we inaugurate a series of individual reviews of expert system shells. In "Elements of Expert System Shells" (p. 62), technical editor Maxine Fontana and associate editor Jordene Zeimetz present the *PC Tech Journal* philosophy and criteria that will be used to evaluate each shell's knowledge representation method, inference engine, control strategies, user interface, and program interfaces. Each review will emphasize specific applications for which a particular expert system shell is best suited. Reviewers will build a different sample application for each shell.

The first review in this series features a powerful rule- and frame-based

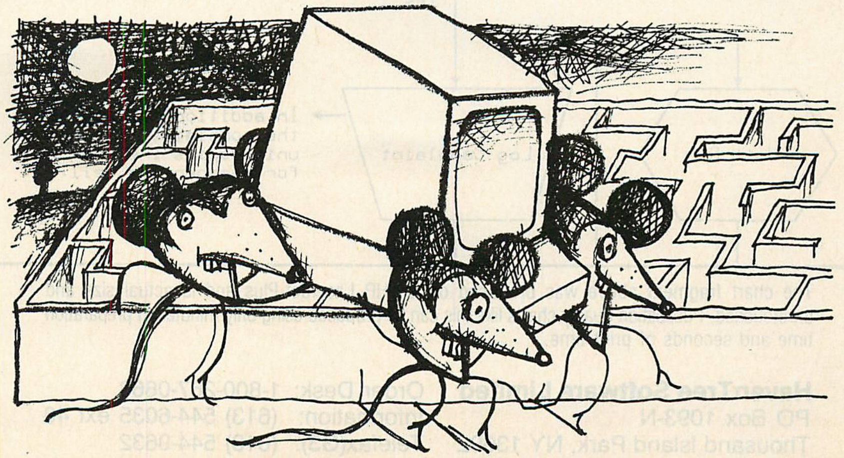


ILLUSTRATION • MACIEK ALBRECHT

expert system shell, GoldWorks from Gold Hill Computers. In "The Age of GoldWorks" (p. 68), author Ken Levine presents a sample expert system that can be used to configure computer systems. Levine heads up an AI software development company.

FIRING UP C CONTROVERSY

In the arena of more traditional application development, the question of which development language is best has ignited the fires of our readers. At no other time in the history of *PC Tech Journal* have our readers been so diametrically opposed and so religiously aligned in their beliefs.

The fire was first kindled with a letter from Wilson Jones in the November 1987 issue, in which Mr. Jones stated, "I fail to see any benefits to programmers coding application programs in C," and claimed that he preferred BASIC over C for performance and maintainability. In February, the fire was fueled with two responses to Mr. Jones's letter. In one, J.R. McLemore defended BASIC, and in the other, Robert D. Wilson stood by C. In that same issue, editorial director Will Fastie expounded on "The Trouble with C" in New Directions, while in our cover suite we reviewed nearly a dozen professional C compilers.

Also in that issue, our reader opinion card asked readers at large, "Does C meet your needs as a professional programming language?" The fire is brought under control in this month's Professional Viewpoint (p. 186), where Jordene Zeimetz reports on the results of that survey in which the proponents of C outnumber its critics by more than 2 to 1.

WHAT'S YOUR PROBLEM?

Our reader opinion card this month, which is bound in front of this editorial, asks for your single biggest software development problem. My guess is that many of you share the difficulties of the industry vendors in figuring out what the end user needs. Others may have problems achieving acceptable performance. Still others may have nightmares about supporting multiple operating environments or multiple hardware platforms.

Whatever happens to be your biggest software development problem, we would appreciate hearing about it. You may find that you are not alone in your opinions, and your response helps us to plan editorial content that directly addresses your needs.

ANNOUNCING ORACLE® FOR 1-2-3. JUST IN TIME - $199

Spreadsheet growing too big and complex? You need a database. No time to learn a database? You need the ORACLE database add-in for Lotus 1-2-3.

If you have Lotus 1-2-3 and $199, you can now solve the six biggest spreadsheet problems:

1 Has your spreadsheet grown so complex you can't keep track of the formulas any more?

2 Have you had to break down your large spreadsheet into many smaller ones?

3 Do you have to manually re-enter data that's duplicated in several spreadsheets?

4 Do you have to manually manipulate rows into meaningful groups?

5 Is recalculation time for seldom-used reference variables eating you alive?

6 Do you wish you could simultaneously share spreadsheet data with other PCs, as well as with minis and mainframes?

Now, ORACLE for 1-2-3 turns your spreadsheet into the world-class database you already know how to use. And without learning a new database language, you can use the very same ORACLE that's the most requested DBMS by minicomputer and mainframe users. All for only $199.

ORACLE for 1-2-3 lets data relationships replace ever more complex spreadsheet formulas. No more time is wasted recalculating seldom-used reference cells. In short, ORACLE puts data back where it belongs. In a database.

With simple extensions to existing Lotus menus, ORACLE for 1-2-3 lets you create new database tables right from rows and columns in your spreadsheet. When you query the database from a cell, you immediately see current database information. Update a spreadsheet cell, and the database is simultaneously updated. A range of cells in your 1-2-3 spreadsheet could really be a window into an online micro, mini or mainframe database anywhere in the world. It's network ready, from LAN to WAN. All as easy as… 1-2-3.

ORACLE FOR 1-2-3: THE NEW STANDARD

ORACLE is the number one database for mainframes, minicomputers and workstations. Software Digest recently rated ORACLE the most powerful and versatile relational DBMS for the PC. And ORACLE is based on SQL, the data management standard endorsed by IBM, ANSI, ISO and the federal government. Now, there is a new standard: ORACLE for 1-2-3. It has the simplicity you've always had, with the power you've always wanted.

THE ADVANTAGES OF DATABASE TECHNOLOGY

If your data is in a database, your spreadsheet only has to deal with the data you're interested in. Which means spreadsheet performance is dramatically improved. But this is just the beginning.

Multiple users can share the same data. Different users can have different levels of security. Mainframe-class data integrity and error recovery mean never having to say "Ooops!"

- CREATE YOUR DATABASE FROM SPREADSHEET COLUMNS AND ROWS. Familiar menus and context-sensitive help guide you through the process.
- QUERY YOUR DATABASE AS EASY AS 1-2-3. Have the query built for you, or write SQL queries yourself. Learn why SQL is the industry data management standard.
- UPDATE YOUR DATABASE AS YOU UPDATE SPREADSHEET CELLS. In update mode, changes in your spreadsheet become changes in your database. And if you make a mistake, you can…
- COMMIT OR UNDO CHANGES. Finally, an "UNDO" command for 1-2-3! When your database changes are complete, you can COMMIT them, or ROLLBACK your database *and* your spreadsheet.
- AROUND YOUR OFFICE OR AROUND THE WORLD, ORACLE for 1-2-3 is network ready for data distribution on LANs and WANs. So data on PCs, workstations, minis and even on mainframes appears as if it's on your local hard disk.

HEARD ENOUGH?

WE'LL PAY SALES TAX AND SHIPPING IF YOU FILL OUT AND MAIL THE ATTACHED COUPON, OR CALL TODAY. YOU CAN ALSO SEE A DEMONSTRATION AT YOUR FAVORITE COMPUTER STORE.

Dear Oracle, **ORACLE for 1-2-3** • Oracle Corporation
20 Davis Drive • Belmont, CA 94002

Yes, I have an ☑ 80286/80386 PC running ☑ DOS 3.0+ and ☑ Lotus release 2.01. I also have ☑ 640KB of RAM, plus either ☑ 1MB of extended memory or I'll reassign 1MB of my expanded (above-board) memory as extended memory. Please send me the database I already know how to use: The ORACLE Database Add-in for Lotus 1-2-3. Enclosed is my ☐ check, or ☐ VISA/ ☐ MC/ ☐ AmEx credit card authorization for $199. Hurry.

Print Name ____ Date ____

Company ____ Title ____

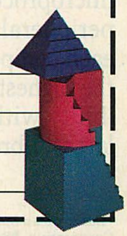
Street (No PO boxes, please) ____

City ____ State ____ Zip ____

Phone ____

Credit Card Number ____ Card Expiration Date ____

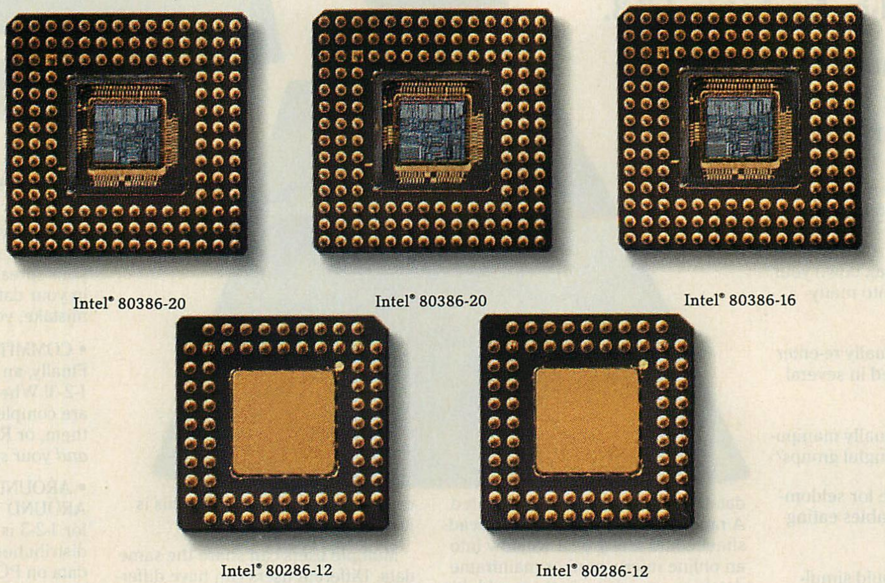
Signature ____ PCTJ



ORACLE®
COMPATIBILITY · PORTABILITY · CONNECTABILITY

CALL 1-800-ORACLE1 EXT. 148 TODAY.

These give you high-performance personal computers.



COMPAQ personal computers offer far more than advanced, high-speed microprocessors. Each offers a combination of innovative features which work with the processor to maximize overall system performance. So there's nothing to slow you down.

Take system architecture, for example. The COMPAQ DESKPRO 386/20 and COMPAQ PORTABLE 386 are built around an advanced 32-bit concurrent bus architecture which exploits the speed of the computers' 20-MHz 80386 microprocessors. Two buses—one for memory and one for peripherals—eliminate information bottlenecks, allowing each component to run at its maximum speed. This ensures the highest system performance without sacrificing compatibility with industry-standard hardware and the world's largest library of business productivity software.

Similar performance enhancements are engineered into each subsystem of every COMPAQ personal computer. Each component is then optimized individually, yet designed to work as part of the total system.

For instance, COMPAQ Fixed Disk Drives deliver both high capacity and high performance. You can install up to a 300-megabyte fixed disk drive in the COMPAQ DESKPRO 386/20 and up to a 100-megabyte drive in the COMPAQ PORTABLE 386. More importantly, you can get to that data almost instantly thanks to some of the industry's fastest access times—averaging less than 30 milliseconds. When you combine this speed and capacity with disk caching, the result is the highest-performance storage subsystem in the industry. To take it one step further, Compaq helps you protect that data with internal high-

*Based on an independent study of major brands. COMPAQ®, COMPAQ DESKPRO 386® and COMPAQ DESKPRO 286® are registered trademarks of Compaq Computer Corporation. ®Registered U.S. Patent and Trademark Office. COMPAQ DESKPRO 386/20™, COMPAQ PORTABLE 386™ and COMPAQ PORTABLE III™ are trademarks of Compaq Computer Corporation. IBM® is a registered trademark and IBM PS/2™ is a trademark of International Business Machines Corporation. Intel® is a registered trademark of Intel Corporation.

THE PROGRAMMER'S SHOP

helps save time, money, and cut frustrations. Compare, evaluate, and find products.

FREE Catalogs and Guarantee

"Whether you're searching for an obscure product no one seems to know about, or you just want to know which of 5 competitors makes the most well-regarded product, our catalogs make finding software easier.

- Comprehensive Product Listing, filled with over 1,000 products.
- Popular Programmer's Tools, containing the most-requested titles (over 300 in all) from each category.
- Dbase Programmer's Catalog with over 60 development tools.

We'll also help you select products with free advice or literature. Plus full guarantee on any recommended product.

Call to request a catalog or information today.

Our Services:

- International Sales Desk
- Compare Products
- Help find a Publisher
- Evaluation Literature FREE
- Programmer's Update
- Dealers Inquiry
- Newsletter
- Rush Order
- Over 700 products
- National Accounts Center

AI TOOLS

Choose from a wide range of tools designed to make AI applications powerful Expert Systems development easy.

LISP and Prolog programmers: choose from many powerful environments in addition to those listed here, or take advantage of the special prices.

Call a Tech Rep TODAY!

Order before 5/31/88 and mention "PT588" for these Special Prices:

	List	Normal	SPECIAL
1st-Class Plus	$495	$399	$359
Arity Expert System Dev't.	$295	$229	$209
Arity Prolog Comp/Interp.	$650	$569	$509
Exsys Expert System Dev't Package	$395	$289	$259
TransLISP PLUS w/runtime	$345	$309	$ 99
Prolog 86 + & Active Prolog Tutor	$315	$278	$229

C Language-Compilers

AZTEC C86 - Commercial	PC	$499
C86 PLUS - by CI	MS	$359
Datalight Optimum - C	MS	$ 99
High C Optimizing Compiler	PC	$549
Instant-C/16M	PC	Call
Lattice C - from Lattice	MS	$259
Microsoft C 5.0 - Codeview		Special*
Microsoft Quick C		Special*
Turbo C by Borland	PC	$ 67

C Language-Interpreters

C-terp by Gimpel - full K & R	MS	$219
C Trainer - by Catalytix	PC	$ 89
Interactive C by IMPACC Assoc.	PC	$189
Run/C Professional	MS	$145
Run/C Lite	MS	$ 79
Turbo C-terp	PC	$119

C Libraries-Files

BTree - Source, no royalties	MS	$ 69
CBTREE - Source, no royalties	MS	$ 99
ctree by Faircom - no royalties	MS	$315
rtree - report generation	PC	$239
dB2C Toolkit V2.0	MS	$249
dbVISTA - Source		Special*
dBx - translator	MS	$299

C Libraries-General

Blackstar C Function Library	PC	$ 99
C Tools Plus - V5.0	PC	$ 99
C Utilities by Essential	PC	$119
Greenleaf C Sampler	PC	$ 69
Greenleaf Functions	PC	$129
LIGHT TOOLS by Blaise	PC	$ 69
Turbo C Tools by Blaise	PC	$ 99

C-Screens, Windows, Graphics

C Worthy Interface Library	PC	$249
dBASE Graphics for C	PC	$ 69
ESSENTIAL GRAPHICS - fast	PC	$185
GraphiC - new color version	PC	$279
Greenleaf Data Windows	PC	$155
w/source	PC	$259
Terminal Mapping System	PC	$279
TurboWINDOW/C - for Turbo C	PC	$ 75
View Manager by Blaise	PC	$199
Vitamin C - screen I/O	PC	$159
VC Screen	PC	$ 79
Windows for C - fast	PC	Call
Windows for Data - validation	PC	Call
ZView - screen generator	MS	$139

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. Formats: 3" laptop now available, plus 200 others. UPS surface shipping add $3/item.

RECENT DISCOVERY

Foxbase PLUS by Fox Software. DBMS conforms to standard MAC interface. Up to 9 definable output windows, definable buttons, import/export hypercard stacks, MAC, Developer Kit, $339

DataBase & File Management

Advanced Revelation	PC	$ 779
Clarion DBMS	PC, List	$ 695
CQL - Query System - source	PC	$ 359
DataFlex by Data Access	PC	$ 595
DataFlex multiuser	PC	$1049
Magic PC	PC	$ 169
Paradox - original	PC	$ 369
Paradox V2.0 List: $725	PC	$ 499

DBASE Language

Clipper compiler	PC	$ 399
dBASE III Plus	PC	$ 399
dBASE III LANPack	PC	$ 649
DBXL Interpreter	PC	$ 99
FoxBASE + - V2.0	MS	$ 259
Quicksilver Diamond	PC	$ 369

DBASE Support

dAnalyst	PC	$ 219
dBASE Tools for C	PC	$ 65
dBRIEF with BRIEF	PC	Call
dBC III by Lattice	MS	$ 169
dBug III	MS	$ 179
Documentor - dFlow superset	MS	$ 229
FORCE III - Dbase compiler	PC	$ 109
Genifer by Bytel-code generator	MS	$ 279
QuickCode III Plus	MS	$ 189
R&R Report Writer	MS	$ 139
Seek-It - Query-by-example	PC	$ 79
Silver Comm Library	MS	$ 139
Tom Rettig's Library	PS	$ 79
UI Programmer-user interfaces	PC	$ 249

Editors for Programming

BRIEF Programmer's Editor	PC	Call
de - EMACS-style	PC	$ 65
Epsilon - like EMACS	PC	$ 149
ME Macro Editor - Source	PC	$ 79
PC/VI - by Custom Software	MS	$ 109
Personal REXX - V1.6	PC	$ 99
SPF/PC - Version 2.0	PC	$ 179
Vedit PLUS	MS	$ 129

***Mention "Special PT588" and get both a good price and FREE Software!**

Call for a catalog, literature, and *solid* value

800-421-8006

THE PROGRAMMER'S SHOP

Your complete source for software, services and answers

5-P Pond Park Road, Hingham, MA 02043
Mass: 800-442-8070 or 617-740-2510 2/88

RECENT DISCOVERY

C Talk by CNS - Object-oriented C programming. Smalltalk-like message formats, contains encapsulation, messaging, inheritance. MS, Lattice, Turbo, C86. MS $ 129

Fortran & Supporting

50 More: FORTRAN	PC	$ 95
I/O Pro - includes No Limit	PC	$229
MS Fortran - 4.0, full '77		Special *
PC-Fortran Tools-xref, pprint	PC	$165
RM/Fortran	MS	Call
Scientific Subroutines-Matrix	MS	$129

Other Products

ASMLIB - 170 + routines	PC	$125
Back-It by Gazelle	MS	$119
Baler	PC	$459
CO/SESSION - remote access	PC	$229
Dan Bricklin's Demo II	PC	$169
Disk Technician-smart upkeep	PC	$ 89
Fast Back Plus	PC	$149
Flash-Up	PC	$ 69
Interactive Easy Flow V5.0	PC	$125
Link & Locate - Intel tools	MS	$309
Mace Utilities	MS	$ 85
MKS Trilogy	MS	$ 99
PC-Metric - analyze complexity	MS	$ 89
PC/Tools Deluxe-by Custom	PC	$ 69
Plink 86 PLUS - overlays	MS	$275
PVCS - by Polytron		Special *
risC by IMSI - H.A.L.	MS	$ 79
Sapiens Make	MS	$155
Source Print - V3.0	PC	$ 75
Tree Diagrammer	PC	$ 65
Visible Computer: 8088	PC	$ 65
WKS Library by Tenon	PC	$ 79

Translator

dB2C - requires toolkit	MS	$249
RTC PLUS by Cobalt Blue	MS	$379
SofTRAN - Translation Lang.	PC	$349
TP2C	PC	$199
Turbo-to-C-Tools by TGL	PC	$479

Xenix/Unix

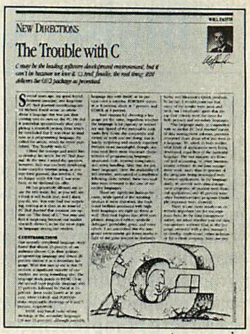
C-Terp by Gimpel Software	$379
Cobol - by Microsoft	Special *
Fortran or Pascal-by Microsoft	Special *
FoxBASE +	$649
Panel Plus	$535
RM/Cobol	$959
Xenix Complete System 286	$999
Xenix Development System 286	$499

FEATURE

Halo '88 - includes scanner control/ scanned image manipulation, 144 devices supported, PS/2. Virtual Raster Interface for EMS. New languages Turbo C, BASIC, MS QuickBasic. PC $229

CIRCLE NO. 162 ON READER SERVICE CARD

LETTERS



CLEARLY C

We were pleased with the review Marty Franz gave the Lattice C compiler in the cover suite on "The State of C" in your February issue ("C Contenders," p. 52). However, we would like to point out three small errors and also let you know of the changes in the compiler since the review appeared.

First, although the text mentions that a librarian is included with the Lattice C compiler, table 1 (p. 56) incorrectly shows that we do not.

Second, Lattice C is available on IBM mainframes and the IBM PC (and compatibles), Commodore Amiga, and Atari systems. Lattice C cross compilers are offered for DEC VAX, Sun, and Apollo systems; we also offer cross compilers from MS-DOS host systems to AmigaDOS, NEC 78312/78310, 68000, and Z-80 systems. But Lattice C is *not* available for the Apple Macintosh, as was stated in the article.

Third, neither Lattice C nor any of the compilers reviewed will allow linking of mixed memory models. However, mixed memory model functions *are* allowed through the NEAR and FAR key words—which is exactly the opposite of what the article stated.

Since the review appeared, Lattice has released version 3.3 of our Lattice C compiler (late March 1988). This new version includes both the MS-DOS C compiler and an OS/2 C compiler. In addition, OS/2 as well as MS-DOS versions of the Lattice Screen Editor (LSE) and the C-SPRITE debugger are now being bundled with the compiler. Other features of version 3.3 include compliance with emerging ANSI C standards, improved embedded system support, enhancements to the standard libraries, as well as advances in the compiler. The price of version 3.3 has been lowered to $450.

Lattice is maintaining a strong commitment to our C compiler and C programming tools. Within the next few months we will introduce a brand new C compiler, portions of which we previewed at COMDEX/Fall '87.

Wayne Nartker
Lattice Inc.
Glen Ellyn, IL

IN DEFENSE OF C

I find "The Trouble With C" misdirected (see New Directions, Will Fastie, February 1988, p. 27). Of the programming languages I know—FORTRAN, PL/1, Pascal, BASIC, REXX, and a few job control languages—I find that C is the best. I like its clean syntax, felicitous balance between freedom and structure, portability, and expressive control structures. I find it easier to write clean, correct code in C than elsewhere because the language makes it easy to say what I mean—very important to good programming.

We have Paradox, and I think of it as highly as Mr. Fastie does. I would never do a Paradox-type application in C; however, I am currently working on a program to compute scale reliabilities using tetrachoric correlations. For this, Paradox somehow does not seem a natural choice.

Mr. Fastie's criticisms are properly directed toward procedural programming languages in general, not to C specifically. Surely high-level development tools should be used on jobs for which they are well suited. My unscientific impression is that resistance is strongest in mainframe COBOL shops, not among PC application developers. But why obscure the issue with an attack on C?

Jonathan Goldberg
Washington University Medical School
St. Louis, MO

I quite agree that choosing the right tool for the job is not only appropriate, but necessary; I, too, have used C when it was best suited for the task. And it is likely that an attack on C could easily be converted into an attack on Pascal or another high-level language, as Mr. Goldberg suggests.

However, my main point is not that C is bad but that for a wide range of business problems, development environments other than C offer significant advantage, most particularly the reduction of development time. Just as I would not attempt to build an operating system with interpreted BASIC, I would prefer to build business applications with an environment that deals with the problem at a higher level than C. Paradox's Personal Programmer (PPROG) and the recently released Clarion Professional Developer are excellent steps in that direction.

—WF

C PERSPECTIVE

The letter from Wilson Jones in the November 1987 issue caught my attention (see "Doesn't C," Letters, p. 13). His situation is similar to mine, but I would focus on the same issue from a somewhat different standpoint. He is saying that the C language does not have the utility for him that it appears to have throughout the industry.

In the world of PCs, I suppose 90 percent or more of all code executed comes from some standard application package, which is called "canned software" in the world of minicomputers and mainframes. On the larger computers, aside from the system software, the majority of application programs are custom written for the user.

When you are building a package for widespread use, it is easy to see how the greater capabilities of C are important. However, Mr. Jones and I might be looking at a small business that has a particular need for software that is central to the operation of the business, yet is not efficiently (from the user's point of view) implemented in a commercially available package. So we will propose to design and code some-

thing that calls for a minimum of keystrokes by the user, has few frills, and provides the information that is needed. It should be straightforward, bullet-proof, and allow for growth in size and scope.

We estimate cost by the hours required, and we want to bring the application up as quickly as possible. We document our application and realize that it may be maintained by ourselves or by others. The cost of development is a prime consideration. In this environment, C doesn't appear to be the right tool for us. Languages such as COBOL, RPG, and BASIC are more appropriate, although we lament certain aspects of each.

Bill Graham
Hawi, HI

GET BACK TO BASICS?

I hope your magazine will consider doing a feature article on the "new wave" BASIC programming languages—True BASIC 2.01, Turbo BASIC 1.1, and QuickBASIC 4.0. All three products support many of the newly adopted ANSI standards for the BASIC language.

Last September, your magazine had an excellent article written by Justin Crom comparing Turbo BASIC and QuickBASIC ("BASIC Face-off," p. 136). Since then, Microsoft has released version 4.0 of QuickBASIC with a brand new compiler featuring a threaded p-code that offers an improved user interface and debugging facilities. Also, QuickBASIC now has program modules similar to those offered by version 2.01 of True BASIC.

In my opinion, magazine editors have focused too much attention on the exotic programming languages such as LISP, FORTH, and C. They seem to forget that BASIC is still the most popular and widely used programming language. In fact, it is the only language many people know. In general, users are more likely to upgrade from a BASIC interpreter to one of the "new wave" BASIC products than they are to switch to an entirely new language such as Pascal or C. Thus, I think most readers would like to read about the exciting new developments in BASIC.

Alan F. Tomala
St. Clair Shores, MI

We have covered BASIC in the past, and we will continue to do so in the future. We think highly of the "new wave" products and certainly agree that they are important for the future.

While Mr. Tomala's statement about the popularity and widespread use of BASIC is accurate, one other category of software is used more frequently than even BASIC or C for applications development: data manager languages. In our data management study, 61 percent of respondents said they used data management software to develop applications for their own company and 74 percent used it to develop applications for their clients. In addition, 55 percent said they also used BASIC, while 38 percent used C and 37 percent used COBOL; the numbers drop sharply after that.

As for the latest version of Microsoft's QuickBASIC, turn to this month's Product Watch (p. 149) for a review of version 4.0.

—WF

INTERFACING TO THE FUTURE

After reading the informative issue of March 1988, I can only say that if the OS/2 Presentation Manager represents intuitive simplicity, heaven help us. OS/2 will make the "how to" book publishers rich.

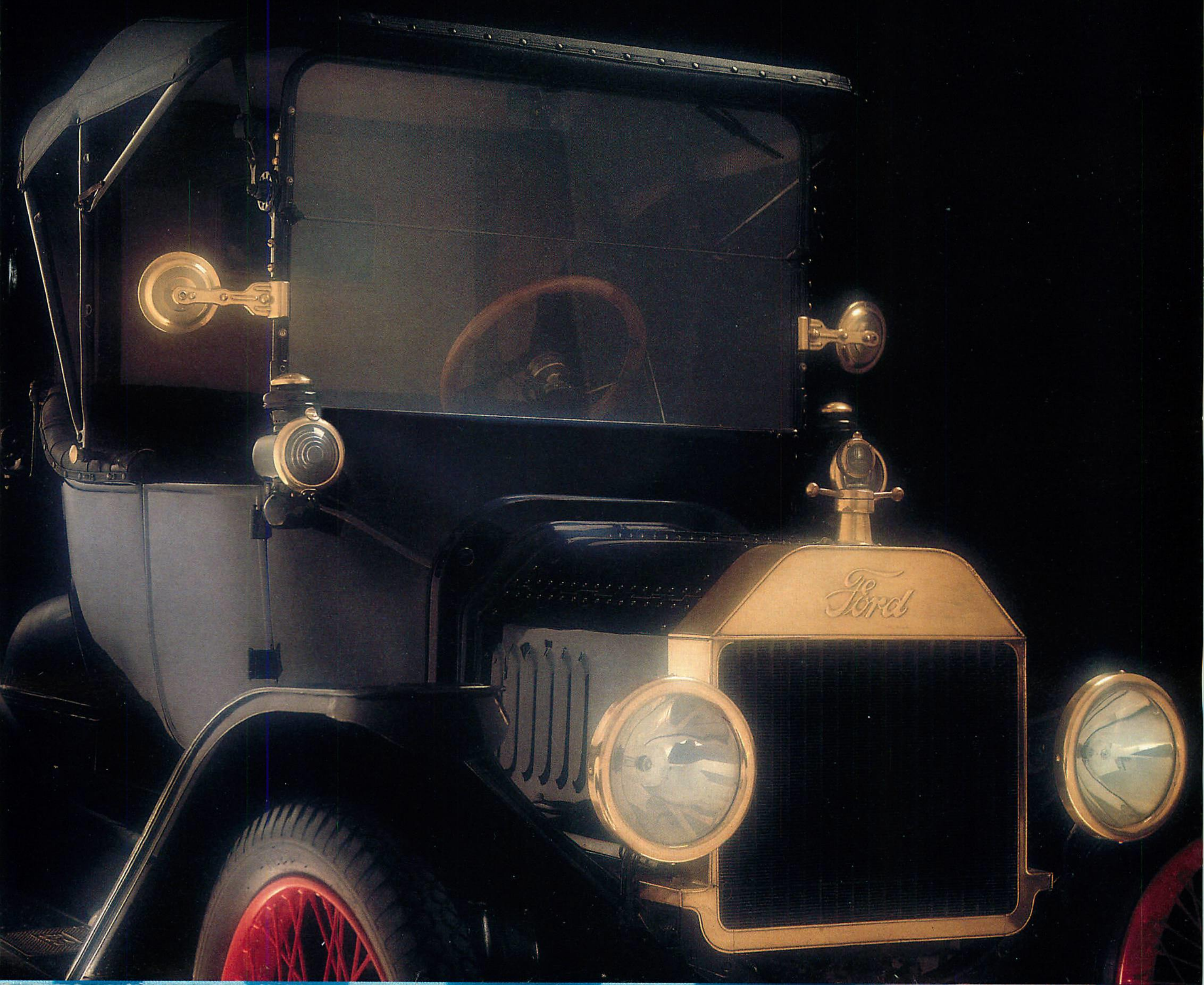
The biggest laugh of all is the notion that the new interface will represent a standard we can be sure will be with us for a long time. Only one thing is for sure; three years from now we'll all be learning something else.

Robert J. Spear
Accokeek, MD

I enjoy *PC Tech Journal* and find it to be a pleasant yet technically informative way to maintain my long-standing experience with the DOS world. However, as one who is concurrently a Macintosh user, I could not help being amused that the March 1988 OS/2 Presentation Manager articles barely mentioned, let alone compared, this competitor who led microcomputing into the user-driven graphic interface some five years before Presentation Manager will be available on the market. Both the Apple and the IBM camps owe a debt to Xerox PARC, of course, but Apple has refined the desktop metaphor to such a high art that Microsoft and IBM have been hard pressed to come close.

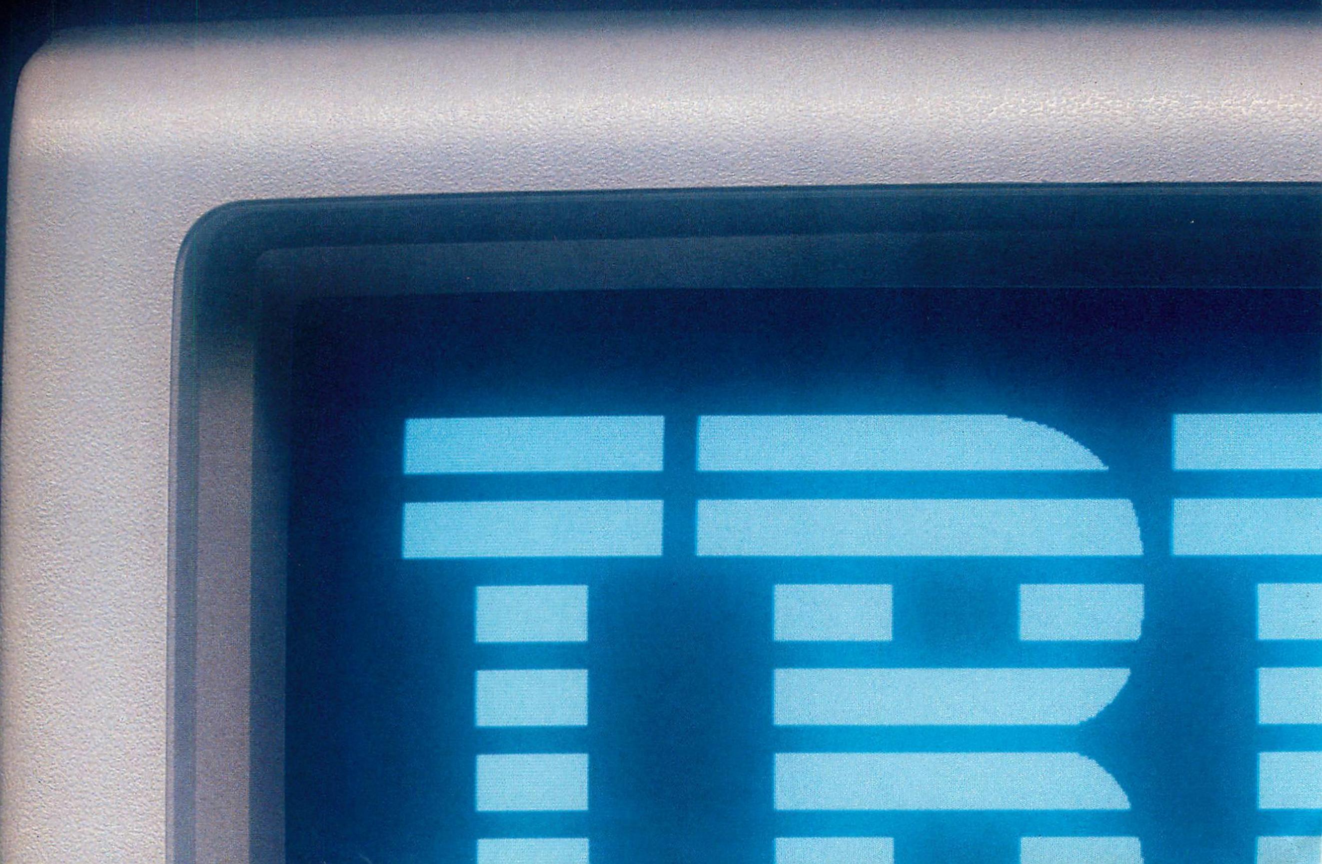
Apart from two tables full of memory-intensive keystroke combinations (the likes of which are becoming too common in the Mac world, too, as DOS applications are ported over), and the unnecessary accommodation of *three*-button mice, the best of what was described was rather similar to the

It Used To Be You Could Have Any Color You Wanted… As Long As It Was Black.



It Used To Be You Could Have Any Personal Computer Solution You Wanted...

You've come a long way since your first 64K personal computer. And the industry has changed dramatically. You've learned about performance, compatibility and connectivity. You've tried to keep everything in balance—vendor to vendor—but you're looking for a better way. And as always, the



As Long As It Was Blue.

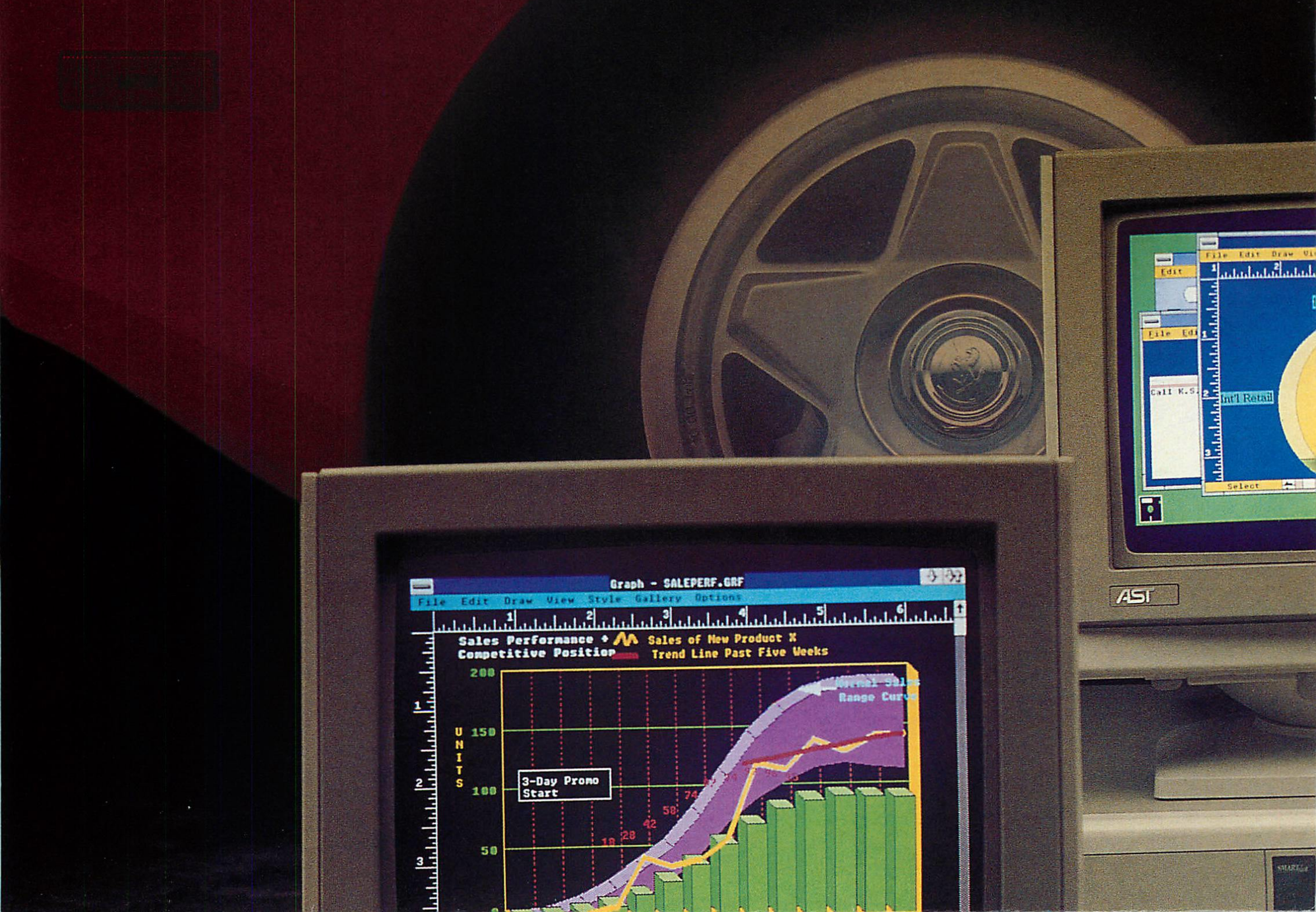
future poses the biggest issues.

When choosing the best and most complete solutions, you've learned the company has as much to do with your decision as the product. And that company should service a wide range of your needs. While it used to be IBM®'s big picture was the only show in town…

Times Cha

It's time to exercise your freedom of choice. With AST.

We've been involved in the PC business almost as long as Big Blue itself. And like you, we know no single product is a solution. That's why we provide a wide selection of system, enhancement and data communica-



s Have nged.

tions products that work as well together as they work by themselves. Products that provide superior performance, room for the future and compatibility with countless other industry products—even our competitors'. Now there's finally a personal computer company that gives you a better alternative.

12 13 14 15 16 17 18 19 20 21 22 23 24 25
Select
AST
AST
AST-VGA
AST offers a complete family of high-performance solutions for **DESKTOP COMPUTING:** Innovative 20
and TurboLaser®/PS. **ENHANCEMENTS:** Multifunction, memory, graphics and system boards for almos
intosh™ II. **CONNECTIVITY:** Complete solutions for linking PCs/XT™s/AT®s, even PS/2s to mainframes

MHz Premium™/386, Premium/286, small footprint Premium Workstations™
t any personal computer, including IBM's PS/2™ series and Apple®'s Mac-
or minis, local area networking and Premium Connectivity Workstations℠.

AST®
RESEARCH INC.

Solutions, Service And Support For Changing Times.

First and foremost, you're searching for solutions. AST has a wide range of them, making us a strong alternative in these changing times. We've been a part of your life, and nearly three million others, from our first PC enhancement cards and connectivity solutions...to those for IBM's PS/2 and Apple's Macintosh II. And over 100,000 of you are already using AST Premium desktop computers.

PC MAGAZINE EDITOR'S CHOICE

1987 PC WORLD WORLD CLASS AWARD

At AST, we know the solution you select has as much to do with the company delivering it...as it has to do with the product. Quality and compatibility are AST traditions. Our service network is worldwide. And your future is built into every product we create. Now more than ever, in every area of desktop computing, it's AST. Because times have changed.

THE BEST OF 1987 PC Magazine

For more information call **(714) 863-9991** and ask for operator number AP1. Or send in the coupon—or a photo copy—today.

☐ Please send me a copy of your new full line Product Catalog.

☐ Please have an AST representative call me.

Name ______________________

Title ______________________

Company ______________________

Address ______________________

City/State/Zip ______________________

Phone ______________________

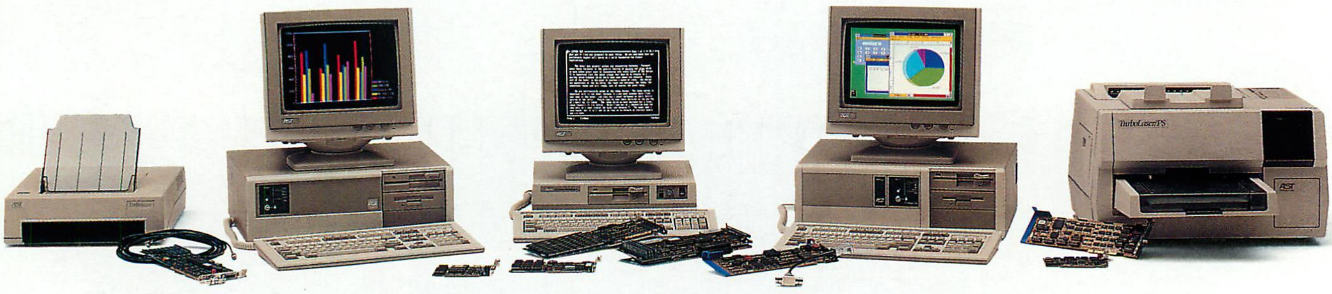
Please list magazine in which this ad appeared:

____

AST Research, Inc. 2121 Alton Avenue, Irvine, CA 92714-4992 Attn: M.C.

AST® RESEARCH INC.

Times Have Changed.



AST markets products worldwide—in Europe and the Middle East call: 44 1 568 4350; in the Far East call: 852 5 717223; in Canada call: 416 826 7514. *AST, AST logo and TurboLaser registered and AST Premium, AST Premium Workstation and AST Premium Connectivity Workstation trademarks of AST Research, Inc. IBM and AT registered and PS/2 and XT trademarks of International Business Machines Corp. Apple registered and Macintosh trademark of Apple Computer Inc.*

good reading found in Apple Computer's *Human Interface Guidelines: The Apple Desktop Interface* (Addison-Wesley, 1987). I can strongly recommend this book to programmers of the OS/2 Presentation Manager as well as to those who work in the Macintosh environment.

As a PC user and programmer who has been won over to the Macintosh for most daily computing activities, I am happy to see that such a software environment is becoming available to the majority of microcomputer users. *PC Tech Journal* would serve its readers well not to ignore the vast experience that has accumulated in five years of programming for the Macintosh—experience that can greatly benefit the OS/2 community directly.

Boyd C. Paulson Jr.
Stanford University
Stanford, CA

Mr. Paulson is right to criticize our Presentation Manager coverage in that it did not compare and contrast the Presentation Manager and Macintosh interfaces. However, the similarities or differences between the interfaces are not as critical as how these interfaces are used.

I agree with him that PC developers could learn much about program design from studying applications residing on the Macintosh. On the Mac, graphics are integral to the presentation of data. On the PC, however, except for paint and business presentation graphics programs, most of the programs we have seen so far that are written to the Windows/Presentation Manager interface tend merely to use the pull-down menus rather than provide a graphical representation of data. This gives operational consistency from application to application but falls short of the true benefits a graphics interface can bring to the user.

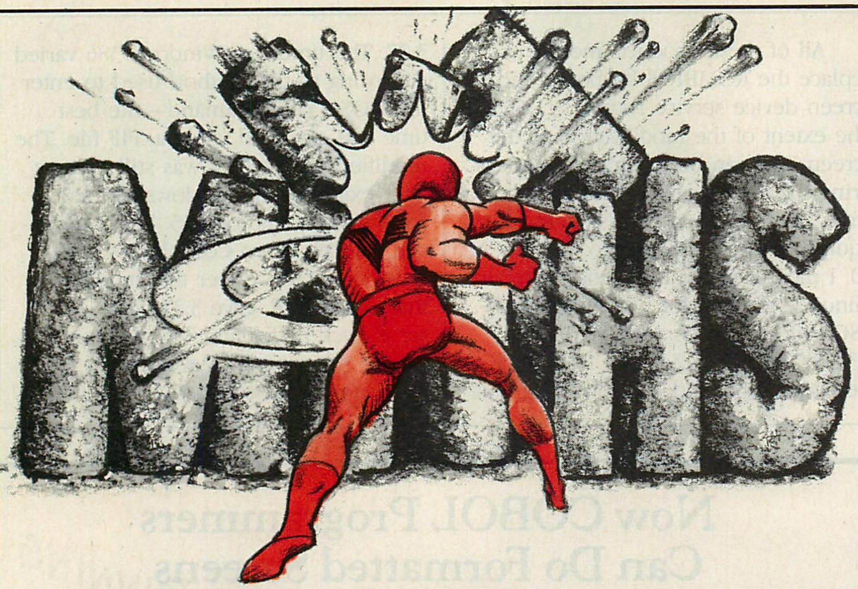
—JA

FANCY DEVELOPMENT

Ed McNierney reviewed The Software Link's PC-MOS/386, Quarterdeck's DESQview, and Microsoft's Windows/386 in "386 Operating Environments" (January 1988, p. 60). They all claim to provide a DOS-compatible environment and, generally, are compatible with the environment provided by DOS as it comes from the box. As soon as goodies such as command-line processors, UNIX-like shells, and screen enhancement utilities, are added, the compatibility isn't so good.

All of these products modify or replace the ROMBIOS keyboard and screen device service routines (DSR). The extent of the modification to the screen DSR can be illustrated by measuring the time it takes to write (type) a large file to the screen. Using an EGA-equipped Compaq Deskpro 386 model 40, I measured the following times (in minutes:seconds) for writing a 350KB ASCII file to screen: DOS 3.31 took 5:36, PC-MOS/386 2.1 took 2:55, DESQview took 2:34, and Windows/386 took 3:27. The times for Windows/386 varied according to the method used to enter the DOS TYPE command—the best time was obtained using a .PIF file. The modified screen DSR was still present after exiting from Windows/386 and gave a write time of 2:55. For comparison, Hersey Micro Consulting's FANSI-CONSOLE, a DOS driver that modifies the screen DSR, gave 3:27 with a scroll-back buffer and 3:16 without.

In my program development, I regularly use FANSI-CONSOLE's scroll-back



buffer feature to review screen output immediately prior to a program failure. I have been disappointed to find that I cannot use this scroll-back driver with PC-MOS/386, Windows/386, or DESQview. Windows and DESQview patch out FANSI-CONSOLE when their screen write routine is installed. PC-MOS/386 will not install a DOS driver. I have tried using Opt-Tech's Scroll & Recall (SR.COM) and Microhelp's utility (TMU.COM), which provide a scroll-back capability using terminate-and-stay-resident routines (TSR). TMU.COM did give DOS-compatible operation in the DESQview environment, but not in PC-MOS and Windows. I could not get SR.COM to operate satisfactorily in any of the environments.

Many TSR programs are sensitive to the order of installation when they are used in conjunction with other TSR programs. These three products actually provide modified DOS operating environments. Some TSR programs can only successfully install into an unmodified copy of DOS. I suspect that one of the most prevalent complaints during the transition to OS/2 will be the failure of favorite TSR programs to install into OS/2's DOS compatibility box.

David L. Spooner
Wilmington, DE

Being compatible with DOS and actually being DOS are indeed not the same thing, particularly where DOS add-on products are concerned. Developing add-on products for operating systems and operating environments that manage the display and keyboard as system resources that are used by multiple tasks is not a simple undertaking; nonetheless, Hersey Micro Consulting and others are investigating providing scroll-back support for DESQview as well as for OS/2.

—JS

MEMORY AT A PREMIUM

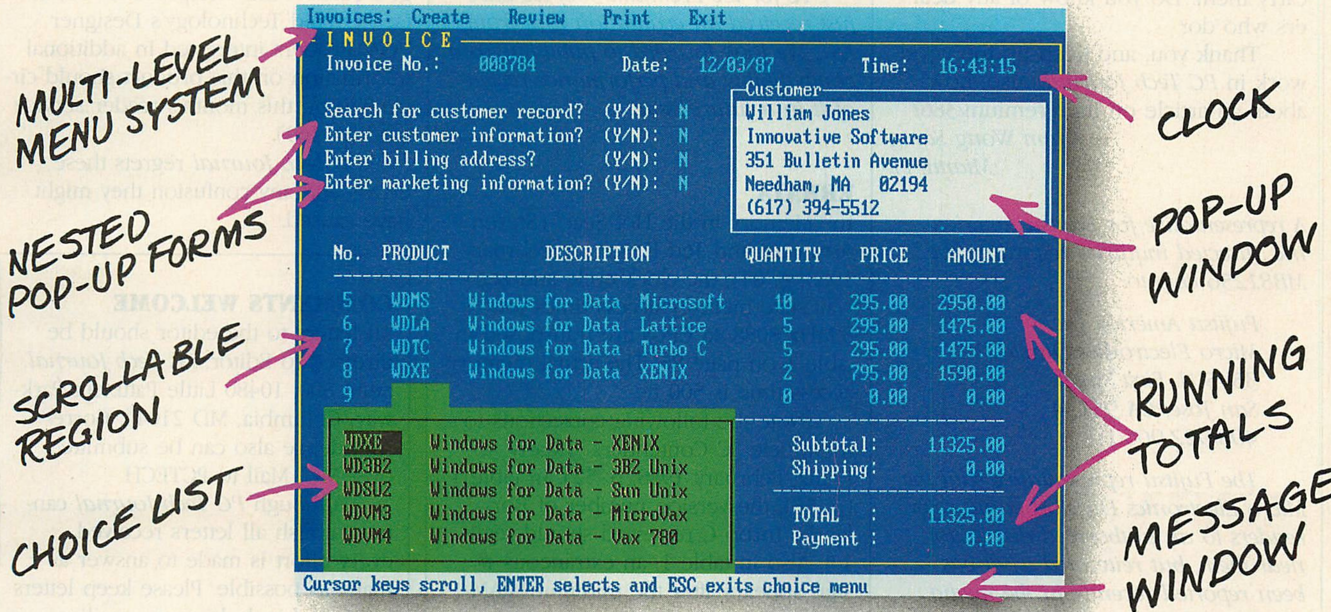
Your article in June 1987 "Compatibility and Performance: Premium/286" (Steven Armbrust and Ted Forgeron, p. 74) was the best review of AST's Premium/286. I had been interested in a 286 for some time, but after I read your review of the machine, I was sold.

I intend to get the model 80 and upgrade it to my own configuration. In doing so, I plan to populate the FASTRAM board myself thereby increasing my RAM and also saving a few dollars. My problem is this: in your article you said, "The FASTRAM card requires 256KB RAM chips with an access time

New Prices

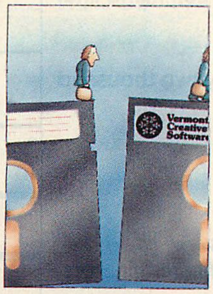
OS/2

WINDOWS FOR DATA®



If you program in C, take a few moments to learn how Windows for Data can help you build a state-of-the-art user interface.

- ☑ **C**reate and manage menus, data-entry forms, context-sensitive help, and text displays — all within windows.
- ☑ **D**evelop window-based OS/2 programs right now, without the headaches of learning OS/2 screen management. Run the same source code in PCDOS and OS/2 protected mode.
- ☑ **B**uild a better front end for any DBMS that has a C-language interface (most popular ones do).



FROM END TO BEGINNING

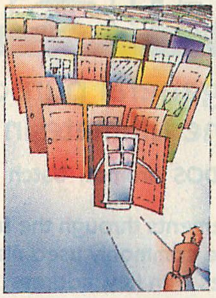
Windows for Data begins where other screen packages end, with special features like nested pop-up forms and menus, field entry from lists of choices, scrollable regions for the entry of variable numbers of line items, and an exclusive built-in debugging system.

NO WALLS

If you've been frustrated by the limitations of other screen utilities, don't be discouraged. You won't run into walls with Windows for Data. Our customers repeatedly tell us how they've used our system in ways we never imagined — but which we anticipated by designing Windows for Data for unprecedented adapatability. You will be amazed at what you can do with Windows for Data.

YOU ARE ALWAYS IN CHARGE

Control functions that you write and attach to fields and/or keys can read, compare, validate, and change the data values in all fields of the form. Upon entry or exit from any field, control functions can call up subsidiary forms and menus, change the active field, exit or abort the form, perform almost any task you can imagine.



OUR WINDOWS WILL OPEN DOORS

Our windows will open doors to new markets for your software. High-performance, source-code-compatible versions of Windows for Data are now available for **PCDOS, OS/2, XENIX, UNIX,** and **VMS.** PCDOS versions are fully compatible with Microsoft Windows. **No royalties.**

MONEY BACK GUARANTEE

You owe it to yourself and your programs to try Windows for Data. If not satisfied, you can return it for a full refund.

Prices: PCDOS $295, Source $295. OS/2 $495. XENIX $795. UNIX, VMS, please call.

Call: (802) 848-7731 ext. 21

Telex: 510-601-4160 VCSOFT FAX 802-848-3502

Vermont Creative Software

21 Elm Ave.
Richford, VT 05476

of 100 nanoseconds. Currently, AST has qualified only Fujitsu MB81256 and Micron MT1259-10 chips for use in the FASTRAM cards." These chips are hard to come by since most dealers don't carry them. Do you know of any dealers who do?

Thank you, and keep up the good work in *PC Tech Journal*. Also, how about an article on the Premium/386?

Adam Wong Samg
Miami, FL

A representative for Fujitsu America Inc. directed inquiries regarding the MB81256 chip to:

Fujitsu America Inc.
Micro Electronics Division
3545 N. First Street
San Jose, CA 95134
408/922-9000

The Fujitsu representative said the Micro Electronics Division could refer readers to an authorized distributor near them, but reiterated what has been reported recently in the media about the shortage of memory chips, including the Fujitsu MB81256.

To obtain the name of the authorized distributor in your area that carries the Micron MT1259-10, contact:

Micron Technology
2805 E. Columbia Road
Boise, ID 83706
208/386-3800

As for the Premium/386, we have just received an evaluation unit from AST. We look forward to publishing our compatibility and performance review of it in a future issue.

—JA

ERRATA

In "Memory in the Hot Seat" (Steven Armbrust and Ted Forgeron, February 1988, p. 84), the clock cycle and zero-wait-state memory access times for 8-MHz 8088/86 CPUs are transposed in table 1 on page 86. The actual memory access time is 500 ns.

Note the following corrections to the article "C Contenders" (Marty Franz, February 1988, p. 52). In table 1 (p. 56), the version number of Borland's Turbo C reviewed should read 1.5. Also in table 1, an extraneous # sign appears after the minimum RAM requirement for the Manx Aztec C compiler. In table 4 (p. 62), the footnote superscript on Microsoft's numerical error rating should reference *f* (Compiler bug, see text), instead of *e*.

In the March 1988 issue, one of the *PC Tech Journal* System Builder contest prizes voted as part of the "Ultimate PC System" was incorrectly identified on page 37 as Orchid Technology's Turbo VGA. The product was, in fact, Orchid Technology's Designer VGA. Readers interested in additional information on this product should circle 340 on this month's reader service card (p. 169).

PC Tech Journal regrets these errors and any confusion they might have caused.

COMMENTS WELCOME

All letters to the editor should be directed to Editor, *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. Correspondence also can be submitted over MCI Mail to PCTECH.

Although *PC Tech Journal* cannot publish all letters received, every effort is made to answer as many as possible. Please keep letters brief, and include name, mailing address, and telephone number. When a letter is lengthy, a diskette is appreciated.

Upgrade Your Technology

We're Programmer's Connection, the leading independent dealer of quality programmer's development tools for IBM personal computers and compatibles. We can help you upgrade your programming technology with some of the best software tools available.

Comprehensive Buyer's Guide. The CONNECTION, our new Buyers Guide, contains prices and up-to-date descriptions of *over 600 programmer's development tools by over 200 manufacturers*. Each description covers major product features as well as special requirements, version numbers, diskette sizes, and guarantees.

How to Get Your FREE Copy: 1) Use the reader service card provided by this journal; 2) Mail us a card or letter with your name and address; or 3) Call one of our convenient toll free telephone numbers.

If you haven't yet received your copy of the Programmer's Connection Buyer's Guide, act now. Upgrading your programming technology could be one of the wisest and most profitable decisions you'll ever make.

USA........ 800-336-1166

Canada.................................800-225-1166
Ohio & Alaska (Collect) 216-494-3781
International.................................216-494-3781
TELEX..9102406879
FAX 216-494-5260

Business Hours: 8:30 AM to 8:00 PM EST Monday through Friday
Prices, Terms and Conditions are subject to change.
Copyright 1988 Programmer's Connection Incorporated

PROGRAMMER'S QUALITY INTEGRITY Service SUPPORT PRICE CONNECTION

Established 1984

ORDERING INFORMATION

FREE SHIPPING. Orders within the USA (including Alaska & Hawaii) are shipped FREE via UPS. Call for express shipping rates.

NO CREDIT CARD CHARGE. VISA, MasterCard and Discover Card are accepted at no extra cost. Your card is charged when your order is shipped. Mail orders please include expiration date and authorized signature.

NO COD OR PO FEE. CODs and Purchase Orders are accepted at no extra cost. No personal checks are accepted on COD orders. POs with net 30-day terms (with initial minimum order of $100) are available to qualified US accounts only.

NO SALES TAX. Orders outside of Ohio are not charged sales tax. Ohio customers please add 5% Ohio tax or provide proof of tax-exemption.

30-DAY GUARANTEE. Most of our products come with a 30-day documentation evaluation period or a 30-day return guarantee. Please note that some manufacturers restrict us from offering guarantees on their products. Call for more information.

SOUND ADVICE. Our knowledgeable technical staff can answer technical questions, assist in comparing products and send you detailed product information tailored to your needs.

INTERNATIONAL ORDERS. Shipping charges for International and Canadian orders are based on the shipping carrier's standard rate. Since rates vary between carriers, please call or write for the exact cost. International orders (except Canada), please include an additional $10 for export preparation. All payments must be made with US funds drawn on a US bank. Please include your telephone number when ordering by mail. Due to government regulations, we cannot ship to all countries.

MAIL ORDERS. Please include your telephone number on all mail orders. Be sure to specify computer, operating system, diskette size, and any applicable compiler or hardware interface(s). Send mail orders to:

Programmer's Connection
Order Processing Department
7249 Whipple Ave NW
North Canton, OH 44720

	List	Ours
386 products		
386 ASM/386 LINK *Cross Asm by Phar Lap*	495	377
386 DEBUG *Cross Debugger by Phar Lap*	195	129
NDP C-386 *by MicroWay*	595	529
NDP ForTran-386 *by MicroWay*	595	529
PC-MOS/386 *by The Software Link All Varieties*	CALL	CALL
blaise products		
ASYNCH MANAGER *Specify C or Pascal*	175	135
C TOOLS PLUS/5.0	129	99
Turbo ASYNCH PLUS/4.0	129	99
Turbo C TOOLS	129	99
Turbo POWER TOOLS PLUS/4.0	129	99
VIEW MANAGER *Specify C or Pascal*	275	199
borland products		
Paradox 1.1 *by Ansa/Borland*	495	359
Paradox 2.0 *by Ansa/Borland*	725	525
Paradox 386 *by Ansa/Borland* ... *New*	895	CALL
Paradox Network Pack *by Ansa/Borland*	995	725
Quattro: The Professional Spreadsheet	247	179
Sidekick Plus ... *New*	200	139
Turbo Basic Compiler	100	68
Turbo Basic Support Products *All Varieties*	100	68
Turbo C Compiler	100	68
Turbo Pascal	100	68
Turbo Pascal Developer's Toolkit	395	285
Turbo Pascal Support Products *All Varieties*	100	68
Turbo Pascal Tutor	70	49
Turbo Prolog Compiler	100	68
Turbo Prolog Toolbox	100	68
Call for other Borland products	100	68
c language		
CBTREE *by Peacock Systems* ... *New*	159	98
CQL *by Machine Independent Software* ... *New*	395	329
Eco-C88 Modeling Compiler *by Ecosoft*	100	69
GraphiC *by Scientific Endeavors*	395	309
Lattice C Compiler *from Lattice* ... *New Version*	450	239
Mark Williams Let's C *with FREE csd*	75	54
WATCOM C6.0 *by Waterloo Computer* ... *New*	295	249
Vitamin C *by Creative Programming*	225	149
VC Screen *Forms Designer*	100	79
creative programming		
Vitamin C	225	149
VC Screen *Forms Designer*	100	79
Reference Database *for Norton Guides* ... *New*	50	45
database management		
Advanced Revelation *by Revelation Tech*	950	659
Clipper *by Nantucket*	695	379
dBASE III Plus *by Ashton-Tate*	695	389
dFLOW *by Wallsoft*	149	119
The Documenter *by Wallsoft*	295	225
FoxBASE+ *by Fox Software*	395	249
FoxBASE+/386 *by Fox Software* ... *New*	595	399

	List	Ours
Genifer *by bytel*	395	249
MAGIC PC *by AKER*	199	167
UI Programmer *by Wallsoft*	295	239
ZIM *by Zanthe Information* ... *New*	880	769
faircom products		
c-tree & r-tree Combo	650	519
c-tree *ISAM File Manager*	395	315
r-tree *Report Generator*	295	239
d-tree ... *New*	CALL	CALL
lahey computer products		
F77L-EM/16 ... *New*	695	CALL
F77L-FORTRAN Compiler ... *New*	477	CALL
Lahey Personal Fortran 77 ... *New*	95	CALL
logitech products		
LOGIMOUSE *All Varieties*	CALL	CALL
LOGITECH Modula-2 Development System	249	199
Modula-2 Compiler Pack	99	75
Modula-2 Toolkit	169	139
LOGITECH Modula-2 Window Pkg	49	39
metaware products		
High C ... *New*	595	529
386 Version ... *New*	895	789
Professional Pascal ... *New*	595	529
386 Version ... *New*	895	789
microsoft products		
Microsoft C Compiler *w/CodeView* ... *New Verison*	450	285
Microsoft FORTRAN *Optimizing Comp* ... New Version	450	285
Microsoft Macro Assembler ... *New Version*	150	99
Microsoft QuickBASIC	99	69
Microsoft QuickC	99	69
Microsoft OS/2 Programmer's Toolkit ... *New*	350	CALL
Microsoft Windows	99	69
Microsoft Windows 386	195	129
Microsoft Windows Development Kit	500	319
Call for other Microsoft products		
mks products		
MKS AWK	75	65
MKS RCS *Rev Control System*	189	155
MKS SQPS *SoftQuad Publishing Software* ... *New*	495	379
MKS Toolkit *with MKS VI*	139	109
MKS Trilogy *with AWK, CRYPT & Korn Shell*	119	99
MKS VI *Editor*	75	65
novell products		
Btrieve *ISAM Mgr with No Royalties*	245	184
Xtrieve *Query Utility*	245	184
Report Option *for Xtrieve*	145	99
Btrieve/N *for Networks*	595	454
Xtrieve/N	595	454
Report Option/N *for Xtrieve/N*	345	269
XQL	795	599

	List	Ours
other products		
Austec/Ryan-McFarland Products	CALL	CALL
Brief *by Solution Systems*	195	CALL
Carbon Copy Plus *by Meridian Technology*	195	135
CCExpress *by Meridian Technology* ... *New*	595	499
Dan Bricklin's Demo II *by Software Garden*	195	155
db/LIB *by AJS Publishing*	139	119
Desqview *from Quarterdeck*	130	105
Epsilon *Emacs-like editor by Lugaru*	195	147
GRASP *Paul Mace Software* ... *New*	99	85
KEDIT *by Mansfield Software*	125	98
Link & Locate *by Systems Software*	350	309
Mace Utilities *Paul Mace Software*	99	85
OPT-Tech Sort *by Opt-Tech Data Proc.*	149	99
OPTASM *by SLR Systems*	195	169
PC Scheme *Lisp by Texas Instruments*	95	77
Peabody *Specify Language, by Copia Intl*	100	89
Periscope *All Varieties*	CALL	CALL
Personal REXX *by Mansfield Software*	125	99
SCREENIO *by Norcom*	400	379
SLATE *by Symmetry Group* ... *New*	299	269
Turbo Programmer *by ASCII* ... *New*	289	229
peter norton products		
Advanced Norton Utilities	150	89
Norton Commander	75	55
Norton Editor ... *New Version*	75	55
Norton Guides	100	65
for OS/2 ... *New*	150	109
Norton Utilities	100	59
program mgmt utilities		
Interactive EASYFLOW *by Haventree*	150	125
PolyMake *UNIX-like Make Facility*	149	129
PVCS Corporate *Version Control System*	395	329
PVCS Personal	149	129
Quilt Computing Products	CALL	CALL
Teamwork/PCSA *by Cadre Technologies*	995	929
TLIB *by Burton Systems Software*	100	89
sco products		
XENIX System V for PS/2 ... *New*	CALL	CALL
XENIX System V 286	1295	979
XENIX System V 386	1495	1145

CALL for Products Not Listed Here

CIRCLE NO. 175 ON READER SERVICE CARD

CALENDAR

MAY

May 5–6
Modeling and Simulation Conference
Pittsburgh, PA (IEEE-CS, ISA, and SCS) *Contact:* William G. Vogt, 348 Benedum Engineering Hall, University of Pittsburgh, Pittsburgh, PA 15261; 412/624-9686

May 9–12
COMDEX/Spring '88
Atlanta, GA (The Interface Group) *Contact:* The Interface Group, Inc., 300 First Avenue, Needham, MA 02194; 617/449-6600

May 15–19
Human Factors in Computing Systems
Washington, DC (ACM SIGCHI) *Contact:* Gail A. Chmura, 5214 Monroe Drive, Springfield, VA 22151; 703/750-9401

May 24–27
Measurement and Modeling of Computer Systems
Santa Fe, NM (ACM SIGMETRICS) *Contact:* Connie Smith, Performance Engineering Services, 1114 Buckman Road, Santa Fe, NM 87501; 505/988-3811

May 31–June 3
National Computer Conference
Los Angeles, CA (AFIPS) *Contact:* American Federation of Information Processing Societies, Preston White Drive, Reston, VA 22091; 703/620-8900

JUNE

June 1–3
National Conference on Management of Data
Chicago, IL (ACM and SIGMOD) *Contact:* Dina Bitton, Dept. of EECS, University of Illinois at Chicago, Box 4348, Chicago, IL 60680; 312/996-0142

June 5–9
Computer Vision and Pattern Recognition
Ann Arbor, MI (IEEE-CS) *Contact:* CVPR 88, c/o The IEEE Computer Society, 1730 Massachusetts Avenue, N.W., Washington, DC 20036-1903; 202/371-0101

June 13–15
***PC Tech Journal* Systems Forum '88**
San Francisco, CA (*PC Tech Journal*) *Contact: PC Tech Journal,* 10480 Little Patuxent Parkway, Columbia, MD 21044; 800/544-7285; 301/740-8300

June 13–17
8th International Conference on Distributed Computing Systems
San Jose, CA (IEEE) *Contact:* Walter Kohler, Dept. of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003; 413/545-0962

June 21–24
USENIX Summer Conference '88
San Francisco, CA (USENIX Association) *Contact:* Judith DesHarnais, USENIX Conference Office, P.O. Box 385, Sunset Beach, CA 90742; 213/592-3243

Attn Btrieve™ programmers:

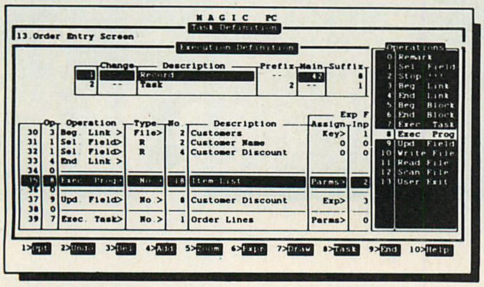
SUDDENLY, MAGIC PC MAKES YOUR DBMS OBSOLETE

You know how database applications are created — by hacking out line after line of time-consuming code. Most DBMS' and 4GL's give you some programming power. But when it comes to serious applications, they keep you bolted to your seat writing mountains of tedious code. And rewriting it all over again with every design change.

Imagine how much faster you'd be if you could replace the painful coding phase with an innovative visual technology which takes only a fraction of the time: Introducing Magic PC—the revolutionary Visual Database Language from Aker Corporation:

- **High-Speed Programming:**

With Magic PC's visual design language you quickly describe your programs in non-procedural Execution Tables. They contain compact programming operations which are executed by Magic PC's runtime engine. You fill-in the tables using a visual interface driven by windows and point-and-shoot menus. One table with 50 operations eliminates writing more than 500 traditional lines of code. Yet with Magic PC you don't sacrifice any power or flexibility.



With a powerful set of high-level non-procedural operations you program at only a fraction of the time.

- **Maximum Power AND Simplicity:**

With Magic PC, you can generate robust DBMS applications including screens, windows, menus, reports, forms, import/export, and much more! Plus, Magic PC has one of the friendliest user interfaces you've ever seen. Using Magic PC you can look-up and transfer data through a powerful Zoom Window system. Magic PC even lets you perform command-free queries.

- **Btrieve Performance:**

Magic PC incorporates Btrieve, the high-performance file manager from SoftCraft. This gives you exceptional access speed, extended data dictionary capabilities, and automatic file recovery!

- **Virtually Maintenance-Free:**

With Magic PC you can modify your application design "on the fly" without any manual maintenance. Magic PC automatically updates your programs and data files on-line! This also makes Magic PC an ideal tool for prototyping complete applications in hours instead of days.

- **FREE Networking:**

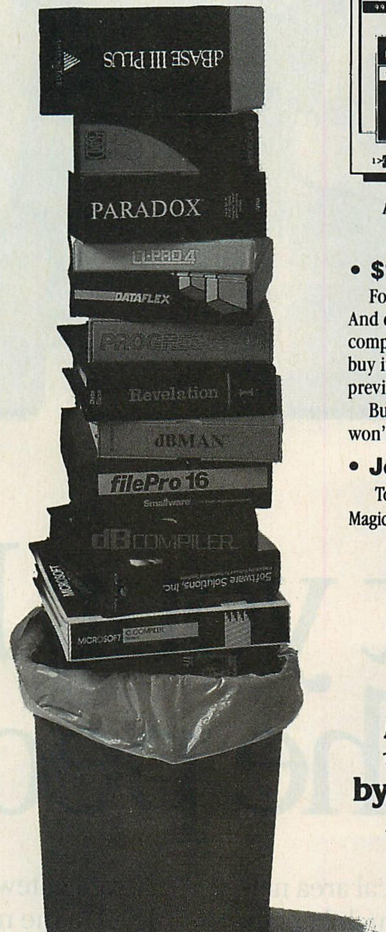
Magic PC comes complete with LAN features. Develop multi-user applications for your LAN with Magic's file and record-locking security levels.

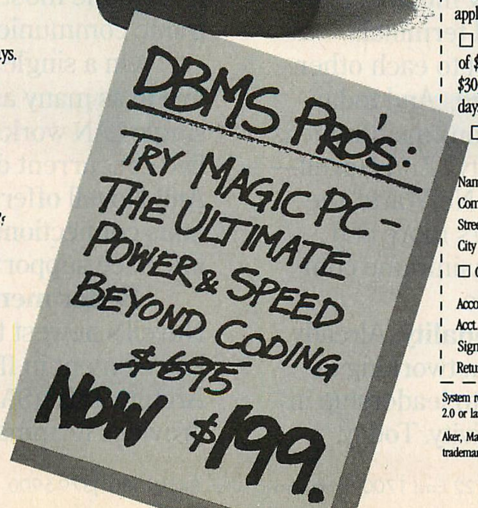
- **Stand-Along Runtime:**

Distribute your applications and protect your design with Magic PC's low cost runtime engine.

- **All For Only $199:**

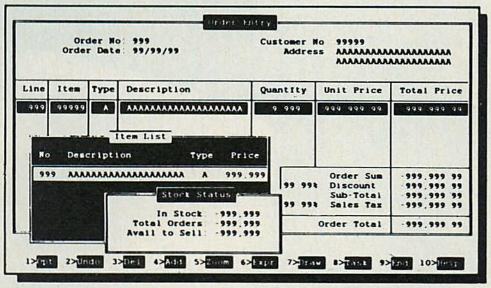
Best of all, Magic PC is an unbeatable bargain. For a limited time, Magic PC's price has been reduced to only $199! Yes, this is the same Magic PC that normally lists for $695! And Magic PC eliminates the need for a separate DBMS, compiler, or application generator. It comes complete with all the tools you need to develop your own database applications instantly.





TECH JOURNAL

"Magic PC's data base engine delivers powerful applications in a fraction of the time. . . there is truly no competitive product."
Victor Wright — PC Tech Journal



Pop-up Zoom Windows run multiple programs per screen — with point-and-shoot data transfer between windows!

- **$199 — With A Money-Back Guarantee!**

For a limited time, you can get Magic PC for only $199. And even at this low price, Magic PC is risk-free. If you're not completely satisfied, simply return it within 30 days and we'll buy it back (less $19.95 restocking fee). And if you'd like a preview, Magic PC's Tutorial Demo is available for just $19.95.

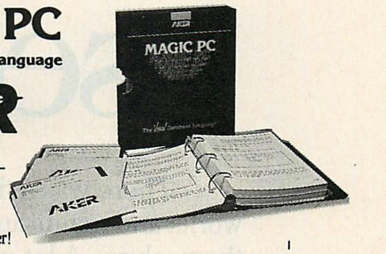
But you'd better hurry — Magic PC's special $199 price won't last long!

- **Join The Magic PC Revolution**

To unleash your DBMS design power, order your $199 copy of Magic PC right now by calling toll-free or returning the coupon below.

ORDER NOW: CALL (800) 345-MAGIC
In CA (714) 250-1718

MAGIC PC
The Visual Database Language
by AKER



Yes! I want to generate powerful applications much faster!

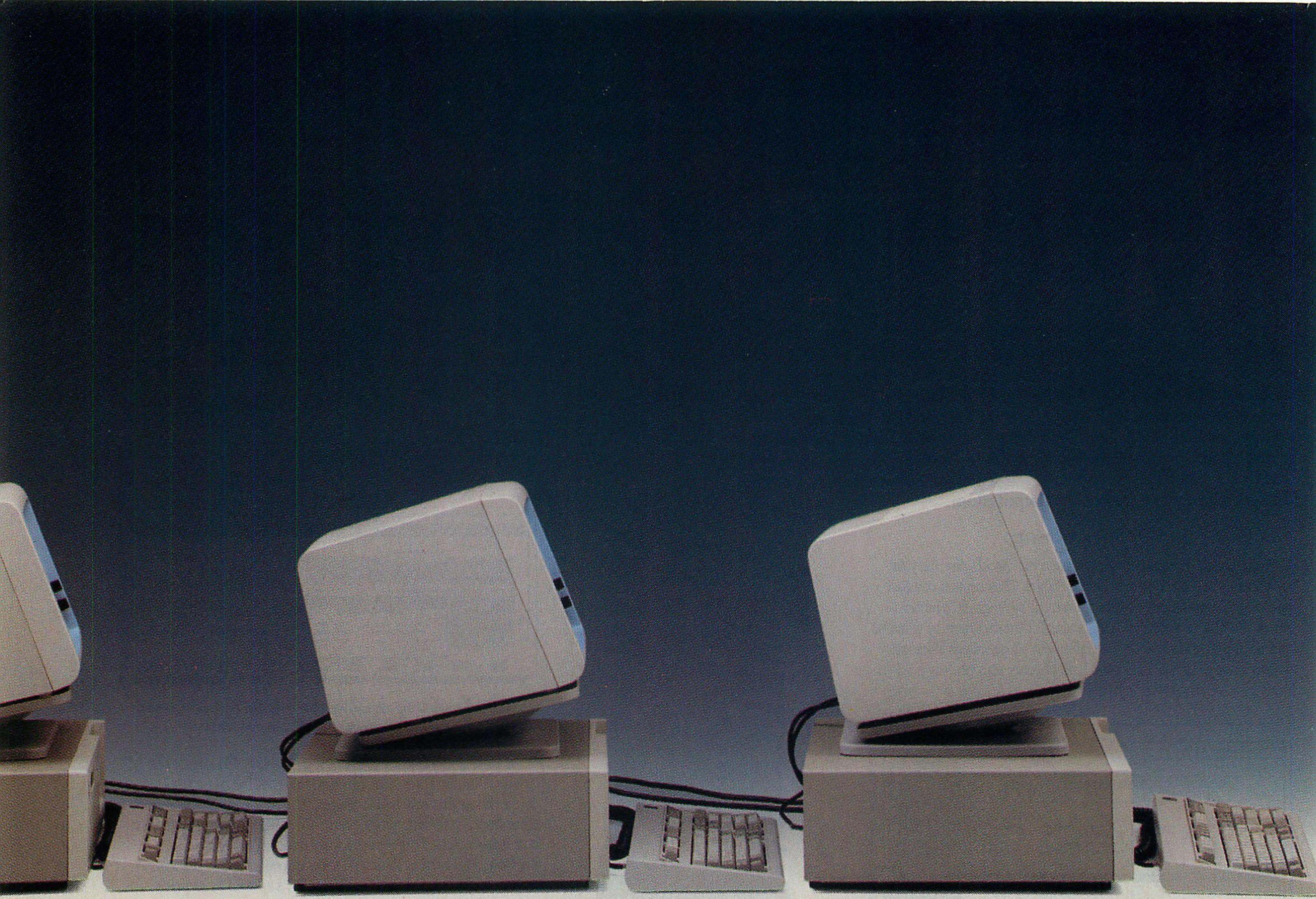
☐ Rush me my copy of Magic PC at the special promotional price of $199 (add $10 P&H, and tax in CA. International orders add $30). I understand I can return Magic PC for a refund within 30 days, if I'm not completely satisfied.*

☐ Rush me a copy of Magic PC Tutorial Demo at $19.95 (add $5 P&H, and tax in CA. International orders add $15).

Name ____________
Company ____________ Phone ____________
Street Address (no POB) ____________
City ____________ State ______ Zip ______
☐ Check enclosed Charge to my: ☐ VISA ☐ MasterCard ☐ American Express
Account No. ____________
Acct. Name ____________ Exp. Date ___/___
Signature ____________
Return to: **Aker Corp.**, 18007 Skypark Cir B2, Irvine, CA 92714

PT DD BS CL DB MS CS CR DP DM A

System requirements: IBM PC, XT, AT, PS/2 or 100% compatible with 512K RAM, hard disk and DOS 2.0 or later. 5¼" format, not copy protected. Dealer pricing available. *Return policy valid in US only.



To get your LAN talk see the people who

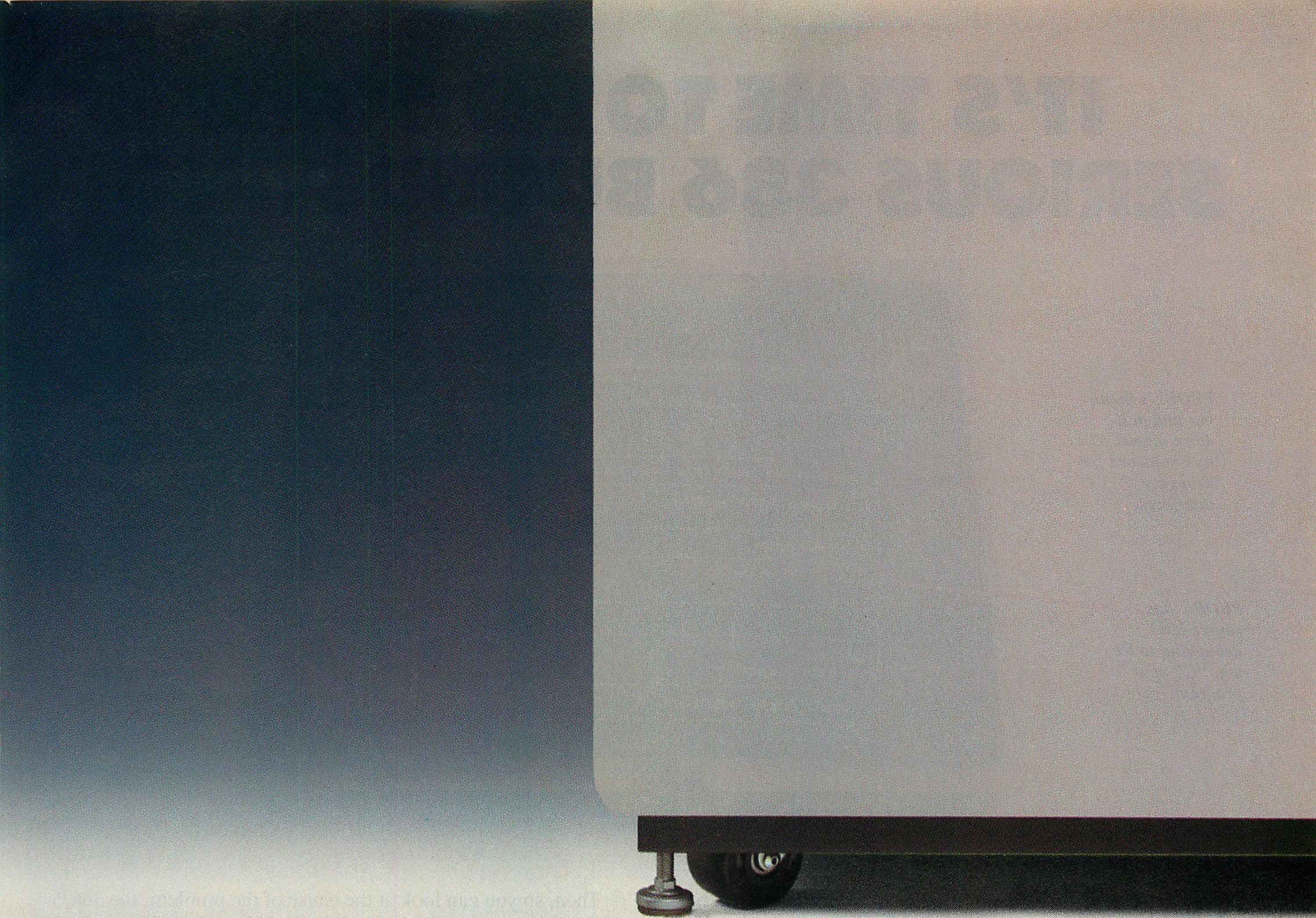
There was a time when local area networks and mainframes pretty much kept to themselves. A lot of PCs and terminals seemed to be talking, just not to each other.

Of course, times change. And today, getting your LAN to talk to your mainframe is not just a notion. It's a priority. And Novell—the company that pioneered NetWare,® the de facto LAN standard—offers more and better solutions for LAN-to-mainframe communications than anyone else.

More gateway functionality. Already the proven innovator in the networking industry, Novell is committed to leadership in micro-to-mainframe connectivity. Today, Novell gateways let you choose from entry-level to the most sophisticated LAN-to-mainframe communications solutions.

On a single gateway you can access from five to as many as 128 host sessions. On a single LAN workstation you can emulate up to five concurrent display or printer sessions. And Novell offers 3270, 5250 and asynchronous connections, as well as IBM Token Ring® interface support.

Better memory utilization with LU6.2. Novell's newest transport protocol is LU6.2, a key element in IBM's Systems Applications Architecture (SAA) strategy. NetWare LU6.2 provides terminal emulation as well as LU6.2



ing to your mainframe, taught LANs to talk.

capabilities from a single workstation with less memory required than any other product available.

Full NetWare integration. All Novell gateways are fully supported by the entire family of NetWare LAN products, including SFT,™ our system fault tolerant operating system. Combining SFT NetWare, LU6.2 and terminal emulation provides a unique solution for network data security and integrity.

If you're ready to start a conversation between your LAN and mainframe, see the company that offers more LAN-to-mainframe products with better flexibility and unsurpassed service. See your Gold Authorized Novell Reseller, or call 1-800-LANKIND.

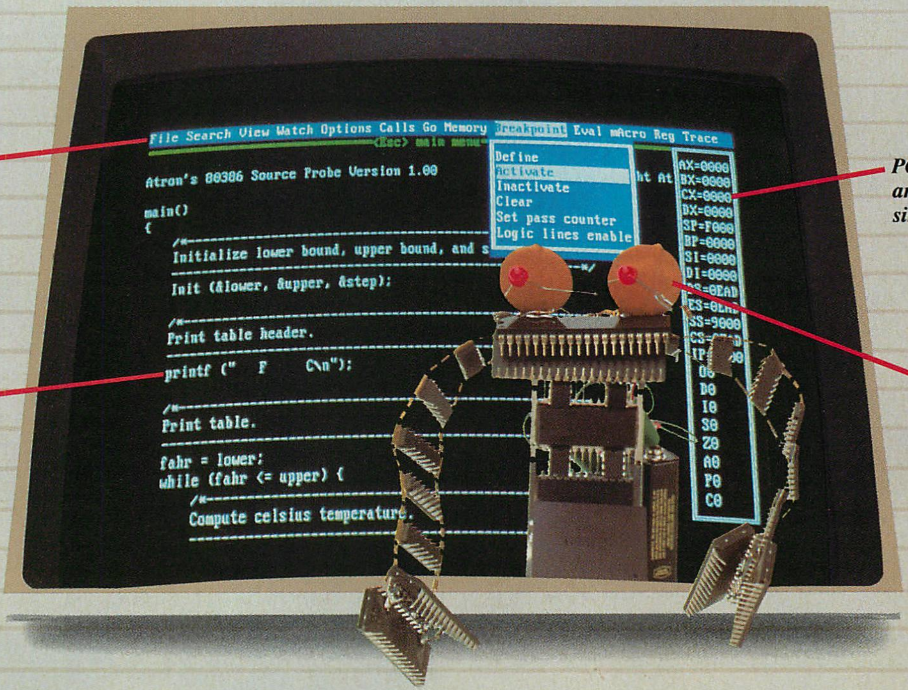
For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVLTM3.

NOVELL

For network solutions, you should be seeing red.

CIRCLE NO. 217 ON READER SERVICE CARD

IT'S TIME TO DO SOME SERIOUS 386 BUGBUSTING!



PROBE's menu bar and pull-down menus set a new standard for debugger interfaces.

PROBE has source-level debugging to let you "C" your program.

POP registers up and down with a single key.

This is an out-of-range memory-overwrite bug. Since it is interrupt related, it only appears in real time.

Welcome to your nightmare. Your company has bet the farm on your product. Your demonstration wowed the operating committee, and beta shipments were out on time. Then wham!

All your beta customers seemed to call on the same day. "Your software is doing some really bizarre things," they say. Your credibility is at stake. Your profits are at stake. Your sanity is at stake.

THIS BUG'S FOR YOU

You rack your brain, trying to figure something out. Is it a random memory overwrite? Or worse, an overwrite to a stack-based local variable? Is it sequence dependent? Or worse, randomly caused by interrupts? Overwritten code? Undocumented "features" in the software you're linking to? And to top it off, your program is too big. The software debugger, your program and it's symbol table can't fit into memory at the same time. Opening a bicycle shop suddenly isn't such a bad idea.

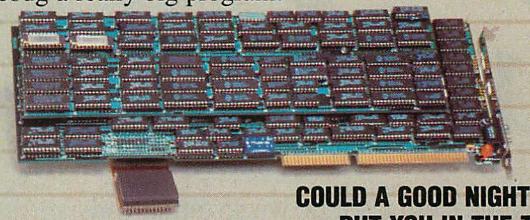
THIS DEBUGGER'S FOR YOU

Announcing the 386 PROBE™ Bugbuster,*from Atron. Nine of the top-ten software developers sleep better at night because of Atron hardware-assisted debuggers. Because they can set real-time breakpoints which instantly detect memory reads and writes.

Now, with the 386 PROBE, you have the capability to set a *qualified breakpoint*, so the breakpoint triggers only if the events are coming from the wrong procedures. So you don't have to be halted by breakpoints from legitimate areas. You can even detect obscure, sequence-dependent problems by stopping a breakpoint only after a specific chain of events has occurred in a specific order.

Then, so you can look at the cause of the problem, the 386 PROBE automatically stores the last 2K cycles of program execution. Although other debuggers may *try* to do the same thing, Atron is the only company in the world to dequeue the pipelined trace data so you can easily understand it.

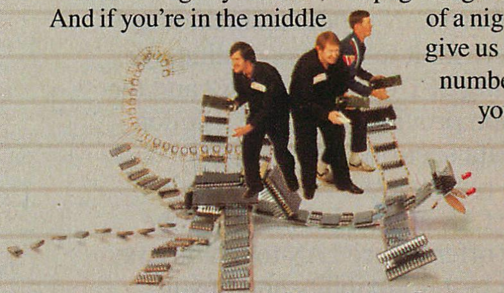
Finally, 386 PROBE's megabyte of hidden, write-protected memory stores your symbol table and debugger. So your bug can't roach the debugger. And so you have room enough to debug a really big program.



COULD A GOOD NIGHT'S SLEEP PUT YOU IN THE TOP TEN?

Look at it this way. Nine of the top-ten software products in any given category were created by Atron customers. Maybe their *edge* is – a good night's sleep.

Call and get your free, 56-page bugbusting bible today. And if you're in the middle of a nightmare right now, give us a purchase order number. We'll FEDEX you a sweet dream.



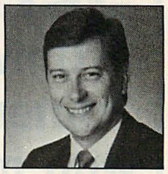
Atron

BUGBUSTERS

A division of Northwest Instrument Systems, Inc.
Saratoga Office Center • 12950 Saratoga Avenue
Saratoga, CA 95070 • **Call 408/253-5933 today.**

*Versions for COMPAQ, PS/2-80s and compatibles. 386 PROBE is a trademark of Atron. Call 44-2-855-888 in the UK and 49-8-985-8020 in West Germany. TRBA

CIRCLE NO. 203 ON READER SERVICE CARD



NEW DIRECTIONS

Distributed Problems

Distributed data and distributed applications are today's buzz words. The possibilities are promising, but the technology is still developing.

The distribution of data and software applications within an organization has become a major focus of attention. This concerns me for a variety of reasons, the most pressing of which is too much hype and not enough substance.

Vendors always seem to know which button to push. Lemon seems to be the scent of choice for today's consumers, but if the next generation prefers cherry blossoms, you can be sure that every dishwashing liquid in the world will switch from "lemon-scented" to the "new, improved fragrance of cherry trees in spring!"

Hardware and software vendors are no different. If distributed this or that seems important, vendors will try to make their products fit the category and call it "distributed."

Although the notion of distributed data and applications is a hot topic, and although many companies say they are pursuing the creation of just such an environment, the available computer science that tells us how to do it is quite limited. That means most solutions proffered by vendors or built internally are ad hoc. We are all still struggling to figure out how this ideal new world can be made to work efficiently, reliably, and safely.

Of the two problems, distributed data and distributed applications, the light is more visible at the end of the applications tunnel. The data management products that run on networks already give us clues about what is required. We can easily see how the same Paradox application, for example, can run on two desktop computers at the same time, connected to its data via a LAN. Borland's Paradox (and other products of the same mold) has two features to make this possible: it uses extensions to the DOS filing system to prevent simultaneous update of the same record, and it communicates with all the other computers on the LAN that are running the same application or are attempting to access the same data. We will come back to the data in a moment; for now, the important point is that the applications are talking to one another.

Getting applications talking to each other in a general way is the major challenge of distributed applications. With a multiuser timesharing system, this was not a problem; the same program ran all the terminals connected to the application, so it could provide the arbitration required to assure reliability. In that situation, the big problem was avoiding deadlocks.

With multiple computers involved, even running the same code, deadlocks are just one of many problems. For example, an end-of-day report might require all transaction activity to finish before it can be run; an interprogram communications channel is necessary to pass such state information between applications. Another problem is that the underlying operating system must be sophisticated enough to supply the features that facilitate such operation; UNIX and OS/2 fit the bill.

No matter what the obstacles, however, distributing applications has tremendous appeal. One of the most compelling reasons to think about moving applications off mainframes or minis and onto the desktop is that the available pool of computing resources is shifting decidedly away from the computer room and onto that desktop. If you will forgive the use of a bad but popular measure, the number of MIPS installed at the desktop now exceeds the number of MIPS installed in all computer rooms worldwide. You can get a rather conservative estimate of this by considering a PC-class machine to have .25 MIPS, an AT-class machine to have 1, and a 386-based machine to have 3. In my own calculation, I liberally conferred 25 MIPS on a 3090 and assumed that all mainframes (even the old 360s) were in that category, which is obviously false.

Most of the desktop power is going to waste. I hate to say this, because I do not want to create an industry slump like we had two years ago when the MIS contingent realized that some of those $5,000 machines were not even being turned on (or couldn't be because they were not hooked up). I fully support buying for the future and think anything less than a 286-based machine today is a bad buy. Nonetheless, vast numbers of machine cycles are going unused out there. What's to become of them?

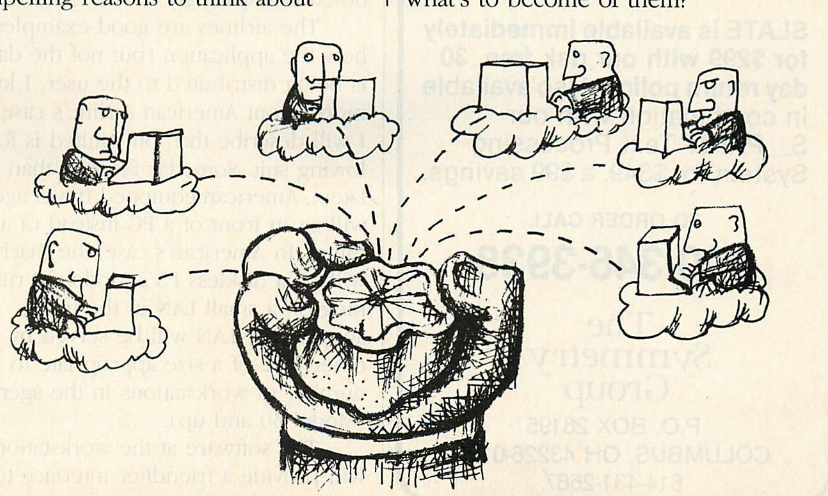


ILLUSTRATION • MACIEK ALBRECHT

It doesn't take long to add up the numbers. The cost of a MIPS at the desktop is almost three orders of magnitude less than the cost of a MIPS at the mainframe and two orders of magnitude less than the mini MIPS. The solution is to get some of those desktop MIPS to take over work the mainframe has been doing at much higher cost and let the mainframe do what it does best, which is handling vast quantities of data.

Although distributing an application is no easy job, it does have an elegance about it. Not only can the mainframe stop running the application code per se, it also can stop running all the support code that handled terminals and formatting and the like. Instead, it can sit in its room waiting for a transaction that is already edited, validated, and formatted into an appropriate structure. The mainframe program then has only to validate the source of the transaction, revalidate the data to assure integrity, and then process them.

The new terminal, a desktop computer, does all that editing and formatting and ships off the neatly packaged transaction. The real benefit is that the application can spend much more of its time working on a transaction. Machine cycles can be allocated to running a beautiful and intuitive user interface, "prevalidating" everything, checking certain data against local storage (for example, there is no branch #77 because this bank has only 30 branches), giving the user help as required, and more. Tens of millions of instructions can be expended on these tasks.

By comparison, spending that kind of time at the mainframe for a single transaction from a single terminal is hideously expensive and wasteful of resource; if mainframe cycles are being shared, you can be sure they are being doled out sparingly.

The airlines are good examples of how the application (but not the data) is being distributed to the user. I know more about American Airline's case, so I will describe that, but United is following suit. Someday (sooner than later), American-equipped travel agents will sit in front of a PC instead of a terminal. In American's case, the machine will be a diskless PS/2 Model 30 running on a small LAN in the travel agency. The LAN will be served by another PS/2 of a size appropriate to the number of workstations in the agency (Model 30 and up).

The software at the workstation will provide a friendlier interface to SABRE, American's reservation system, so that clerks with less training can be productive. Although using the command-line interface is much quicker and more productive, it is also cryptic and therefore prone to error; by contrast, the workstation software takes information gathered from the visual interface and sends off correct transaction requests every time.

Although American will not speak about its very long-range plans, we can imagine further improvements to the interface. A favorite of mine is visual seat selection. Once a flight has been chosen, SABRE would tell the workstation the type of aircraft involved and which seats were taken. The workstation would then pull a graphic representation of the aircraft from its local files (on the LAN server) and present a seating chart to the agent in the form of a picture of the aircraft. The agent would simply point to the seats desired (without even having to know the seat designation), click the mouse, and another properly constructed transaction would zip off to the mainframe.

This simple concept is possible only if the agent is sitting in front of a computer with adequate resources to cope with a graphics user interface. I have seen mainframes brought to their knees with far less; a mainframe could not handle multiple users requiring an intuitive, easy-to-use interface, much less that kind of graphics support.

My imagination roams further. If the front end to SABRE is just a PC at a travel agency, it could really be any PC with the appropriate software installed and a modem attached. It could be the PC my assistant or your secretary has, or even the machines we have at home. American would probably like that most of all; it would not have to pay for the computers installed in a travel agency, while the desktop machine I paid for running on electricity I have to buy is performing useful work that makes money for the airline.

This brings to mind an old joke. Early in this century, the phone company realized that if the rate of new telephone installations continued at the same level (not to mention rising), it would have to turn every American into a telephone operator. So it did: a dial appeared on the phone.

DISTRIBUTED DATA?

I like the American Airlines example for another reason. Although the application is easily distributed with obvious benefits, the data are another story. In

the case of a realtime, transaction-oriented environment (such as an airline reservation system), the data still have to be centralized.

The reason for this is simple and universal. No matter what the application, there can be only one correct answer to any given question at any given time. Think about the question, "How many seats are available on this flight?" Now imagine what would happen if American distributed its data to the 25 largest metropolitan areas and updated its central files every night (as many companies do with sales data, for example). No one would tolerate waiting until tomorrow to find out if a flight was *really* available.

You might say that the logical course of action is to query all the distributed sites on-line. The fact is that today's traveler would not even tolerate *that* much time. The answer to a travel question has to be delivered instantly, sooner if possible. The only way American can manage that is if all the data are in the same place.

That problem is not unique to airlines. In my own 15-year business career, I have seen managements go from accepting monthly sales reports to demanding them daily. As early as 1973 I was building on-demand sales reports into point-of-sale systems. By the turn of the century, the on-line, realtime model will be as much an accepted, standard method of operating a business as batch was in 1980.

These are pressures that force the data to be centralized, not distributed. Distributing the data creates several problems that work against effective transaction processing.

- *Performance.* If data are spread out all over the place, it takes time to find them. The more widespread, the more time it takes. Furthermore, and this is a key point, the underlying systems software must have the facilities to allow a data set to be found no matter where it is placed and no matter where it might later be moved. Adding those capabilities (which are not in place today nor visible in the near future, by the way) will involve additional operating system code and therefore greater overhead.
- *Reliability.* With a centralized data set, you know that all the data are present and correct. If the data are widespread, you not only have to find them, but also have to be *able* to find them (all connections in place, all software up) and know when they have not been found (a LAN server with live data is down or even switched off). Finding all the data requires extensive dictionaries and directories, as well as strict working disciplines and conventions. In the broadest scenario, a new computer should be able to join the application, including creating its own local data files, without disrupting the overall operation of the system or creating live data that other participating applications are unable to find. Creating such an environment is extremely difficult.
- *Maintainability.* Just ask any desktop computing manager in a large company about version control (getting the right update of WordPerfect or Lotus 1-2-3 to everyone who needs it and making sure everyone is compatible) to see how big a headache it is to maintain a huge population of PCs. The solution: centralize the executable version of the programs and load them from the network, not the local hard disk. Now imagine that most of those machines contain live data. Stock up on aspirin; the headaches will get worse.

In the face of all the talk about distributed data, I see a tendency, at least to an extent, to return to centralized data. Many vendor and end-user companies talk about distributed data when they should be talking about distributed applications. For example, workstations running data management software against data stored on the LAN server are not processing distributed data. The data are all in one place on the LAN server, while the application is running on each workstation. This scenario is now very popular because it is solving the problem of having too many data sets on individual workstations—data sets that often contain duplicate data items but, alas, not quite properly duplicated.

Centralizing the data on a server eliminates most of the administrative headaches without taking away what the end user most wants: a highly interactive, user-friendly interface. The user still has a personal desktop computer and all the flexibility it brings, while the company can exercise better control over its valuable data.

A complete recentralization of data is not in the offing, however. Most businesses do not have the requirement for the split-second responsiveness that characterizes an airline reservation system. If it takes a few seconds for a transaction to access several nodes to check various types of data,

and if that process is a valuable one to the company, then several seconds is acceptable. However long it takes, it is certainly going to be much quicker than processing by human hand.

More important, many businesses assign the maintenance of a particular kind of data to a particular group of people. Those are the people who work with it most often and who can most benefit from the best performance their computer systems can deliver. It makes sense for their data to be close (electronically speaking) to them. A data set could be so small that one person could manage it. If that is the case, then the data might reasonably reside on that person's machine.

I am in that situation at the moment. I have a data set that contains the names of the people who have agreed to be panelists for the *PC Tech Journal* Systems Forum. Because it is my responsibility to line up these speakers, no one else needs to see this data set on a regular basis; those that do need to see it will tolerate the updated list I post on the tackboard every few days. However, suppose one of the editors wants to check his or her list of active authors against my list of panelists to see if any match. It would be nice if my data set could be accessed almost as if it were on the querying editor's workstation.

In most businesses, it is exactly this type of ad hoc, "I wonder if . . . " query that leads to the germination and incubation of ideas by gathering and analyzing information outside the normal scope of a person's daily responsibility. This means that cells of live data may spring up around a company in a variety of places, and these data have to be considered as important to the operation of the company as any data stored centrally. An increasing amount of information will be distributed in this ad hoc fashion, and it will only move toward the center as it needs to be available for higher-performance transaction processing.

This is one of the reasons that Structured Query Language (SQL) is getting such attention today. In many respects, SQL is the lemon scent of the computer industry at the moment. But there is no magic to SQL, nor is there any magic associated with data stored in sets accessed via SQL-based queries. The real value of SQL lies in its potential to become the standard communications protocol that allows all data managers to communicate with one another, both to transmit queries (transactions) and results.

This protocol could have a language different from SQL (some people think SQL needs work before it is fully suited to this task). The point is that the data manager on your PC can send a message to a LAN server or a mainframe that will be understood by their respective data engines, and the result will find its way back to the originator of the request. Further, the protocol would have superior transactional capabilities; pieces of the transaction could be synchronized and posted at different data servers to assure proper recording of information and to guarantee a fault-tolerant transaction.

The Sybase/Microsoft/Ashton-Tate SQL Server, and the more recent announcement of the relationship between Lotus and Gupta Technologies, are the first indicators that proper management of distributed data will be possible. The evolution of SQL as the communications glue should guarantee that widely distributed applications can access widely distributed data efficiently, reliably, and safely.

Will Fastie is the editorial director and founding editor of PC Tech Journal.

IRMA + IBM FOR $395

MAINLINK II BY QUADRAM—A BEST BUY

...IN TERMINAL EMULATION

For the first time, you can have terminal emulation that's IRMA®, IRMA 2® *and* IBM® register-compatible. And it costs just $395 per adapter board, or $545 with easy-to-use MainLink II emulation and file-transfer (IND$) software. With MainLink™ II adapters from Quadram you save $750 to $800 compared to IRMA 2 boards.

IBM *AND* IRMA ON ONE BOARD

There's no need to learn new macros or keyboard layout. And the same adapter gives you IRMA plus IBM for much less than either board alone.

Having both standards in one adapter means that you can maintain access to the huge library of applications that were developed for IRMA. And you'll be ready for OS/2™ and the growing list of IBM API-compatible software.

INNOVATIVE, EFFICIENT DESIGN

Here at Quadram, we drew on our years of computer enhancement experience to implement the latest advances in microprocessor and custom chip technology. That let us slash the number of components on the board, cut our costs, and give you access to both major PC-to-mainframe emulation standards at such a modest price. The innovative board also draws less power and runs cooler, which boosts reliability.

MainLink II's half-card size means you can install it in any laptop PC—such as the Datavue® SNAP™—that accepts add-on boards. You can turn that PC into a truly portable terminal. The half card also saves precious space inside the case of desktop and "luggable" PCs.

FUTURE COMPATIBILITY

To assure continued compatibility, MainLink II's design allows for soft-loaded microcode. This means future features and enhancements can be added with just a diskette.

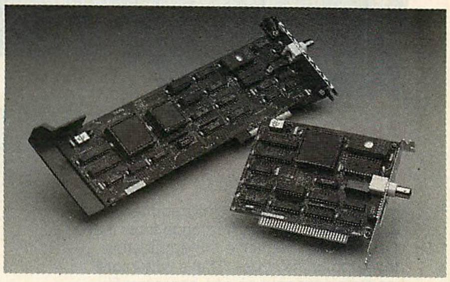
	MainLink II	IRMA2	IBM Ver 2.0/3.0
Devices Emulated	3278/79 3287	3278/79 3287	3278/79 3287
IRMA Compatible	●	●	
IBM Compatible	●		●
Soft Loaded Microcode	●	●	
Power on Self Test	●		
PC/XT/AT/PS2 Compatible	●	●	●
3174, 3274 Supported	●	●	●
Emulation and IBM IND$ File Transfer	●	●	●
Microchannel Available	●	●	●
Price	*$545	#$1,294	+$1,020

* - MLII-E $545.00, MLII-EM Micro Channel Version $645.00 MLII $395.00 - No Software, MLII-M $495.00 - No Software

\# - IRMA2 $1294.00 + $99.00 for IND$ File Transfer Support

\+ - IBM Board is $595.00 and Version 2.0/3.0 Software is $425.00

CIRCLE NO. 172 ON READER SERVICE CARD



NEW VERSION FOR MICROCHANNEL™ BUS

You can even link your new PS/2™ computers into SNA at a great price: just $495 for the Microchannel bus-compatible MainLink II board ($645 with MainLink II emulation and file transfer software) vs. $1,294 for the equivalent IRMA 2® board.

START SAVING NOW

Make MainLink II your standard and watch the savings add up. The next step is for you to assess MainLink II's performance and cost-saving potential. Call us at (404) 564-5666 for dealer information. For product information call (404) 564-5657.

QUADRAM®
An Intelligent Systems Company

One Quad Way, Norcross, GA 30093
(404) 564-5657

PC EXPRESS™ MAKES YOUR "C" PFANTASIES COME TRUE... NEW LOW PRICES!

C/PAC-TWO OF PHOENIX'S MOST POWERFUL C TOOLS FOR ONE LOW PRICE

PforCe

Phoenix has amassed legions of functions into the ultimate C library—over 400 routines which stand at the ready to slash your development schedule and bring in your job ahead of budget.

- Pre-coded optimized object-oriented toolkit
- 400+ routines for C programmers
- Includes databases with B-trees, windows, string handling, menus, interrupt-driven communications and DOS interfaces
- Complete set of low-level hardware interface functions
- Supports Lattice, Microsoft, CI-86 and Turbo C
- Full source code and NO ROYALTIES

Pre-C

Pre-C™ is like UNIX's "lint". It finds problems your compiler won't. Problems that a debugger will have trouble figuring out. It looks at all segments of your program at once and reports inter-module calamities.

- More functionality than UNIX™ LINT™ for C
- Reports incorrect, obsolete and non-portable C usages no compiler would catch
- Improves portability of code
- Available for latest releases of Microsoft, Mark Williams, Lattice, CI-86, and Aztec C compilers
- Additional compiler support can be added by supplying Pre-C with arguments and functions

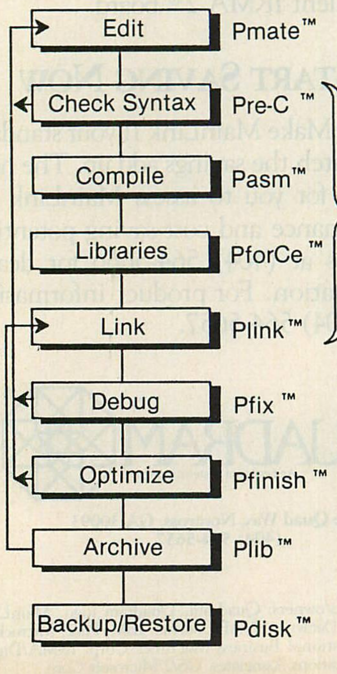
	List:	PC Express
PforCe	$395	$194
Pre-C	$295	$144

C/PAC SPECIAL ONLY $299

PHOENIX PROGRAM DEVELOPMENT TOOLS

Today's professional programmer demands high performance tools that speed up and enhance the application development process. That's why more programmers rely on one company for the best engineered, highest performance tools available—Phoenix.

Phoenix offers a full line of powerful, yet easy-to-use tools that help programmers write, test and deliver more efficiently the best applications possible. Complete, fully detailed documentation accompanying each tool enables quick mastery of the product. In addition, all Phoenix tools are backed by full-time customer support professionals respected throughout the industry. As a result, many Phoenix tools are already established as industry standards and others are fast on their way.



PASM86: *Macro Assembler With Math Co-Processor Code*

High speed, fully MASM-compatible 8086 macro assembler with superior syntax checking and support for 8087, 80286 and 80287 operating code mnemonics. Can define local symbols in the current procedure, assemble files with up to 15,000 symbols, define symbols at assembly time, obtain listings of error and warning lines only. Detailed descriptions and examples of each processor instruction. Now includes Pfix-Lite, a subset of Pfix86 Plus. List: **$195** Ours: **$98**

PFIX86 PLUS: *Multi-Window Symbolic Debugger Does Overlays*

Easy to use, menu driven, multi-windowed symbolic debugger that works with any IBM or Microsoft compiled language. Accesses the full symbol table provided by MS Link or Plink86 Plus. Automatically handles Plink86 Plus-overlaid or resident programs. Source, assembly translations, stack, data areas, and breakpoints displayed simultaneously. Features include: In-line assembler for temporary patches, temporary and permanent breakpoint settings, full speed or trace modes, dual-monitor support, up to 100-step traceback. List: **$395** Ours: **$194**

PFINISH: *Shows Where To Improve Program Speed*

Fine-tunes a software product by identifying inefficient sections of code that need rewriting for maximum performance. It analyzes your program during execution, and snapshots which routines were reached, their callers, how many times each is executed, how much time is spent in each, how many instructions are executed in each. Unlike other "profilers", can use symbol table information to produce much more meaningful analyses on overlays and interrupts. List: **$395** Ours: **$194**

PTEL: *Communications You Can Put On Hold*

Communications for use with most popular modems such as Hayes and compatibles, DEC, Racal Vadic, Anchor, US Robotics and Novation. Ptel automatically adapts to Telink, XModem, Kermit or Modem 7 for CRC checking and for ufn and afn (i.e. "wildcard") file name list transfers, if the bulletin board or the other end computer supports them.

Highly configurable, with choices temporarily or permanently saved. Ptel runs fully interactive or can be batch driven from a script. You can even exit to DOS, move files around or run another application and then return to Ptel, all without dropping the line. List: **$49** Ours: **$39**

PMAKER: *Compile & Link Scripts to Manager Big Jobs*

Similar to the UNIX™ "make" utility. Tell Pmaker all the elements comprising your system and it won't forget. It keeps track of which modules in a program are changed, and recompiles, reassembles, and relinks them to produce a finished product—all with a single command. An essential tool for managing large, complicated, or distributed programming projects. Pmaker works with any compiled language, linker, or other tool you use. List: **$125** Ours: **$69**

PMATE: *Text Editor With Famous Macro Powers*

A full screen, fully customizable text processor/editor with advanced features including: ability to run in the background, C and FORTRAN specific macros, automatic disk buffering, ten individual auxiliary buffers. It is menu, mouse, or command driven with extensive macro command language and a unique last-in, first-out "garbage stack" that saves deleted items for recovery. List: **$195** Ours: **$98**

PFANTASY PAC: *Bundle Up To Save a Bundle*

A super value pac of Phoenix goodies. Includes Plink86 Plus, Pmate, Ptel, Pfix86 Plus, Pmaker and Pfinish. List: **$995** Ours: **$549**

PDISK: *Backup & Disk Management Plus Caching*

Complete disk management package. Cache will significantly speed up disk operation on PC/XT/AT by keeping data in memory instead of disk. It's compatible with the Lotus-Intel-Microsoft (LIM) expanded memory specification, as well as extended memory. Extensive Backup/Restore combinations will include or exclude files, whole and partial subdirectories, and will backup by date/time, file type, or only files changed since last backup. Supports AT high-density floppies, PC floppies, and any storage device accessible through a device driver. List: **$145** Ours: **$89**

PLINK 86 PLUS: *Overlay Linker with Caching Smarts*

Only linkage editor containing advanced overlay capabilities. It handles any compiler or assembler producing standard Intel or Microsoft OBJ files, including COBOL and FORTRAN, Lattice C, CI C-86, Microsoft/IBM languages, and mbp/COBOL. Virtual memory management ensures ample capacity for symbol and common block names (35,000). Plink86 Plus supports unlimited size file, unlimited modules and 4,095 overlays nested 32 deep. Merges object modules, caches overlays in extended or expanded memory, and automatically reloads overlays upon function return. Includes Plib86 library manager. List: **$495** Ours: **$259**

PC EXPRESS™ PROGRAMMER'S WORKBENCH

CLIPPER *From Nantucket*

Compile dBASE for Speed, Protection

Clipper™ turns lumbering dBASE® into a speed dMON with benefits bobbing in its wake: your source code is submerged from public view, you can distribute your compiled application without royalties, and your customers don't even need copies of dBASE! The Summer '87 Clipper offers index files compatible with dBase III Plus, and networking capabilities to run compiled programs on major networks supporting DOS 3.1 with no restrictions on number of users. Clipper offers arrays, menu-building commands, user-defined functions, context-sensitive help techniques for applications, a debugger, and it supports Expanded Memory. It goes well beyond dBASE with 1,024 fields per data base and 2,048 active memory variables.

Clipper has the power to save and restore multiple screens to and from memory variables. You can also create overlays, call object modules compiled in other languages, and create function libraries to link with your applications. Power and flexibility make it the #1 dBASE compiler. List: **$695**, Ours: **Call**.

McMAX *From Nantucket*

Like dBASE for the Macintosh

McMax™ is like running dBASE on the Macintosh. It combines an easy-to-use menu-driven ASSIST mode using the Mac interface, an interactive command mode like dBASE at the dot prompt, and an application programming language fully compatible with dBase III. It gives you the power to create dBASE language applications on the Macintosh and transfer back and forth to the IBM® world. McMax accommodates up to 16 million records, 32,000 characters per record, 255 characters per field, and up to 32 files open concurrently. No copy protection. List: **$295**, Ours: **Call**.

dBASE AT THE SPEED OF C

dBx Translates dBASE Applications to C

You dBASE™ programmers know what an expressive and readable language dBASE is. It's a very comfortable development environment. But the price is debased performance. Even compiled dBASE doesn't offer the speed that some users require these days. The kind of speed offered by software written in the C language. The answer is dBx™.

dBx translates dBASE to C. It offers you a major competitive advantage over the next dBASE programmer: Keep writing in dBASE. Take every application all the way to completion. Then use dBx to translate them top to bottom to C!

Other advantages: C is portable, even to other operating systems like UNIX/Xenix™. To the Macintosh or Amiga. dBx gives your applications a passport to places dBASE cannot go.

Has its own file manager for single user, but links to major C file managers—c-tree and dBC—for compatibility with dBASE files or multi-user support. We have everything you'll need, including good advice.

	List:	PC Express:
dBx	**$ 550**	**$ 419**
with Library Source	**$ 950**	**$ 725**

Call for Full Source Pricing

dBC *Identical dBASE III Plus Files Using C*

dBC™ is a series of C libraries from Lattice which creates, accesses and updates files identical to those of dBASE itself. So dBASE can read and update the files too.

What for? It means both C and dBASE applications can operate on the same data bases interchangably. It means C programmers can interface with the big market of dBASE users out there, yet side-step the dBASE language. It means dBASE applications can now be linked to the universe of C libraries and tools to add windows, graphics, statistical analysis, all the things dBASE cannot do. It means the speed and power of C to impress clients accustomed to dBASE!

dBC's functions parallel all dBASE's file handling commands, many decomposed to permit direct data manipulation. Our versions of dBC mimic file formats for dBASE II and III and now dBASE III Plus makes your programs network ready!... as many stations as a network allows. Hands-off mode handles record and file locking and unlocking automatically. Close in functions give you direct lock/unlock control.

Supports all four memory models. dBASE II, III...List: **$250**, Ours: **$169**. dBASE III Plus...List: **$750**, Ours: **$499**. Call for Source Code Pricing.

THE SPINDRIFT LIBRARY

Fully Functional Fortran Library.

Spindrift's smooth package offers something previously unavailable to the Fortran buffs...a basket of functions, packed to the brim. No more tedious coding. Just call on Spindrift with it's armoury of functions.

Any number of Windows may be defined, each with a striking border and brilliant color. Define "pop-up" screens, Save/Restore images, set Cursor shape. Snare an entire screen into a Character array.

The Keyboard; read it without echo, or sense any keypress during execution. Cursor controlled directly with Edit keys.

DOS interface includes: Call System, Call Exec, Findfrst/Findnext for (*) and (?) file searching. Search the Path, subdirectory and file manipulation, command line argument parsing, date/time functions, DOS environment access, and much more. Other utilities also included.

Specify compiler, Microsoft or Ryan McFarland. List: **$149.00**. Us: **$129.00**

30-DAY MONEY-BACK GUARANTEE

We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out many programs themselves.** Ask your salesperson for details. There's just nothing stopping your buying from PC Express.

C-WORTHY *NEW VERSION!*

INTERFACE LIBRARY

The C-Worthy™ Interface Library wraps an entire user interface around your application. Its full power can be summoned by only a few high level calls. Sound exaggerated? A single function call can set up a complete text editor in a screen window. Recently acquired by Solution System, over 600 pages of Documentation, Turbo and Quick C version and a complete Interface Library have been added.

- High level calls pop menus and scrollable choice lists to the screen, restoring the background when dismissed.
- Windowing facilities open portholes of up to screen size for viewing virtual screens larger than the physical screen.
- Full context-sensitive help screen management takes over these chores and error messages. Automatic routines interrupt with pageable text windows explaining what to do next.

Novell found it "played a key role and accelerated development" in making its NetWare™ utilities easier for users. Ingenious demo: call for it.

Ask for:	List:	PC Express:
C-Worthy	**$195**	**Call**
with Forms Library	**$295**	**Call**

BASTOC *BASIC Into C*

For a trifling price, BASTOC™ moves truckloads of BASIC code over to C. It's a translator which takes in Microsoft Extended BASIC and emits pure K&R C for Microsoft or Lattice. Structures even convoluted BASIC code. Optimized to dramatically reduce execution time. Dynamic string allocation ends BASIC's catatonic halts for garbage collection. Huge worksaver. List: **$495**, Us: **$399**.

BTRIEVE *B-tree File Manager Plus Add Ons*

If networks are on your horizon, betting your future on Btrieve as the one file manager for your C, Pascal, BASIC, and COBOL projects looks like a smart move. Reason? Novell bought Btrieve's creator.

Btrieve's function library takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, space recapture, forward and backward searching. Finds any key in a million in four or less accesses. Files may have up to 24 indexes; fixed record length to 4090 characters; variable length to 64k; indexes up to 255 characters; files of 4 billion bytes.

List: **$245**, Us: **$195**. Network Version List: **$595**, Ours Only: **$465**. Ask about Xtrieve for Btrieve file inquiry, Rtrieve for report writing. For any network that supports the MS-DOS 3.1 file sharing function.

PANEL PLUS

Library Source Code Gives It Complete Portability

There are no end of tools for screen design and data entry, but none quite like Panel Plus. Design a screen under program control, use Panel's utility to "run" and test it field by field, then pass it to Panel's code generator which delivers C source code. Options style the code to your compiler's liking, and you can of course do what you like to the source afterward. The code calls Panel Plus's function library, but now *the library comes in source*, so everything produced is highly portable. Not like other screen managers delivered as object libraries and which leave *you* to write the detailed code.

Panel Plus will operate in graphics mode via interfaces to graphics products it supports and can utilize the EGA's 43-line screen. Low-level I/O functions adapt it to various keyboards, screens, operating systems.

Panel's newest incarnation has every imaginable feature. A single screen design can have 1000 fields stacked as visual overlays up to 127 levels deep or as pop-ups. Groups of fields can be moved between levels. Screens can be output as compilable code or stored on disk for loading at run-time. Each field can be boxed, colored, multi-row, word-wrapped, and scrolled horizontally and vertically if larger than its on-screen view aperture. It can be assigned its own help and error message, can be told to accept certain characters, or to match a picture, and to check data after entry—proper dates, number ranges, etc.—using Panel's or your own validation routines. You can add your routines to Panel's test utility because even it comes as source. Fields are accessed in any order and control reverts to your application program after each field for choice of action.

For past Panelists, the new version has smaller and faster field and screen functions, tighter granularity, and an enhanced, reworked library. Major tool for the serious developer. List: **$495**, PC Express: **$395**.

POLYTRON VERSION CONTROL

Source Code Control for Any Language

PVCS allows programmers, project managers, librarians and system administrators to control the proliferation of revisions and versions of source code in software systems. Independent programmers, the leading software publishers and LAN companies, and hundreds of Fortune 1000 companies rely on PVCS to store and retrieve multiple revisions of text. It maintains a complete history of revisions as an "audit trail", generates status reports, and uses intelligent "difference detection" to minimize disk space for each new version.

On Corporate and Network PVCS simultaneous changes to a module are merged into a single new version. If changes conflict, the user is notified. The "Logfiles" used to track changes are interchangeable between any PVCS product.

Corporate PVCS is for multiple programmers. It includes "branching" to maintain code when programs evolve on multiple paths. Personal PVCS offers most of the power and flexibility of corporate PVCS, but excludes multiple programmer features. Network PVCS is the Corporate version for LANs. File locking and security levels can be tailored to each project.

PolyMake Still using a prehistoric Make? Step up to PolyMake, the most popular and powerful Make utility. Most flexible macros of any Make. Now integrated with PVCS and PolyLibrarian for faster performance and precise configuration management.

Ask for:	List	PC Express:
Personal PVCS	**$149**	**$109**
Corporate PVCS	**$395**	**$309**
Network PVCS	**Call**	**Call**
PolyMake	**$149**	**$109**

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Express, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $12, each add'l $6. UPS Next Day Air and Federal Express shipment costs based on weight. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders or Literature, Call Us Toll Free at...

PC EXPRESS™

A Division of PC BRAND, 150 Fifth Ave., New York, NY 10011

1-800-321-7733

PHONE: (212) 242-3600; FAX: (212) 627-1171;
TELEX: 667962 (SOFT COMM NYK)

PROGRAMMING PRODUCTIVITY TOOLS

BRIEF *Is Anything But. A Whopper of an Editor*

With a name that belies its thoroughness, Brief™ has every feature you've ever contemplated for your editor-in-chief. Text in multiple buffers is scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text blocks may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers, with as many "undo" levels as you want.

Brief has text search abilities rivaling "grep", with wildcards for matching and indifference to intervening characters.

If you use Lattice, C86™, or Wizard, and have 320k, you can compile your C program without ever leaving Brief. It finds the lines with errors, and marches you through the text for repairs.

Parts of Brief were written with its own easy-reading Lisp-like macro language which has structure, conditionals, loops. "Simply the best text editor you can buy", Dvorak *Infoworld*. (Needs 192k.) List: **$195**, PC Express: **Call.**

C-TREE & R-TREE

B-Tree File Manager Now Has Report Generator

	List:	Ours:	Combined
c-tree:	$395	$285	$499
r-tree:	$295	$235	

c-tree: The only major b-tree file manager with network support in the standard low-cost version, c-tree™ gives you record-locking routines for DOS 3.1/3.2, UNIX and XENIX, and it even comes in C source code, yet there are no royalties. Source sticks to K&R, so C-tree is portable. Tests in many environments prove it.

Permits any number of keys for a data file—alpha, numeric, even floating point. Handles files with varied record lengths, multiple keys in one index file. Both high level and decomposed functions. It's the works.

r-tree: Adds the ability to produce ad hoc reports from files maintained by c-tree (v. 4.1 and up). Link a file description to the r-tree™ library, and use any text editor to write report scripts with no further C coding. Reports can access data in several files, select on criteria, join findings into new logical records, sort them, calculate new fields and columns, tabulate by control breaks. Comes in source, same portability as c-tree, and fits any compiler.

WINDOWS for DATA

FIRST PRIZE!

M'Soft Windows Compatible

"Only one package can be easily recommended" said *Computer Language* (June '87) reviewing nine window and data entry products for C. Complete field level functions specify prompt string, field length, data type, screen location, picture, target variable, entry rules, help messages, even functions to call for validation once data keyed in.

Windows for C is a subset. No data entry but all windowing functions. Unlimited windows can be made either to pop up or permanently overwrite the screen, scroll and highlight lists vertically and horizontally. Specify Compiler. Windows for Data: List **$295**, Ours **$259**. Windows for C: List **$195**, Here **$149**.

IN-SYNCH

Remote Control Computing

Control your office, business or any remote computer from your home. Use an inexpensive PC at one location to access a "fully loaded" PC at another. A programmer can monitor and resolve problems on any remote PC from another location....just as if he was sitting in front of that remote computer. Easily and quickly transfer files, spreadsheets, memos...all via your telephone lines (or serial port). Although you need a copy at each station—worry not because for a limited time, PC Express is offering you two for the price of one AT **$295**

Shopping List for the Power Workbench

	LIST	US
ASSEMBLER		
Microsoft Macro Assembler with Utilities	150	109
PASM 86 *by Phoenix, Macro Assembler*	195	98
ASSEMBLER Support		
Btrieve Softcraft's File Manager	245	169
GSS CGI....*Device independent graphics*	495	375
BASIC		
Microsoft BASIC Interpreter..*for XENIX*	350	Call
Microsoft QuickBASIC...*Ver 4.0*	99	Call
Turbo BASIC.......*NEW from Borland*	99	64
BASIC LIBRARIES & UTILITIES		
Btrieve Softcraft's File Manager	245	169
GSS CGI....*Device Independent Library*	495	375
Halo Graphics by Media Cybernetics	300	219
Source Print...*Aldeberan's beautifer*	99	75
Tree Diagrammer..*Also from Aldeberan*	77	67
C LANGUAGE COMPILERS		
C86 PLUS by Computer Innovations	497	397
Lattice C Compiler Now ver 3.2	500	299
Let's C Compiler from Mark Williams Co	75	55
Mark Williams C full development system	495	369
Microsoft C Compiler with free CODEVIEW	450	295
Microsoft QuickC....*Special Price*	99	64
Turbo C.......*New from Borland*	99	64
C LIBRARIES—Communications		
Asynch Manager by Blaise	175	135
Greenleaf Communications	185	139
Essential Communications	185	125
Essential Communications Plus	250	189
C LIBRARIES—FILE MANAGEMENT		
Btrieve Softcraft's File Manager	245	169
Btrieve/N File Management for Networks	595	429
Ctree by Faircom,with full source	395	285
Rtree...*Report Gen. for Ctree*	295	235
Ctree & Rtree.....*Special Combination*	650	499
dBC ISAM Accesses dBase files	250	169
with Source code	500	349
dBC III Plus multiuser	750	499
with Source code	1500	995
Opt Tech Sort Super fast sort for Btrieve	149	105
C LIBRARIES—Graphics		
Essential Graphics..*no royalties*	250	183
GSS CGI....*Device independent graphics*	495	375
GSS Metafile Interpreter stores images	295	235
Halo by Media Cybernetics	300	219
with Dr Halo III......*Paint Utility*	440	299
Halo For Microsoft Languages	595	434
C LIBRARIES—Screen Design		
Curses from Lattice,UNIX lookalike	125	99
with source	250	199
C Worthy....*by Custom Design Systems*	195	Call
C Worthy with Forms	295	Call
Greenleaf Data Windows	225	155
with source	395	249
Microsoft Windows Dev. Toolkit	500	Call
Panel Plus by Roundhill	495	395
View Manager for C,Blaise	275	199
Vitamin C....*Creative Programming*	225	198
VC Screen...*Source code Generator*	100	81
Windows for C	195	149
Windows for Data	295	259
Zview....*Data Management Consultants*	245	175
C UTILITY LIBRARIES		
Basic C....*Basic-like routines for C*	175	139
Blaise C Tools Plus/5.0....*f/MsC & QuickC*	129	99
Blaise Turbo C Tools...*f/TurboC*	129	99
C Food Smorgasbord by Lattice	150	109
C Utility Library by Essential,300 functions	185	119

	LIST	US
Greenleaf Functions	185	139
PforCe by Phoenix, vast library	395	194
OTHER TOOLS		
BASTOC...*JMI,Translates BASIC to C*	495	399
dBX Translator......*dBASE to C translator*	550	419
with Library Source	950	725
Pre/C....*by Phoenix, like UNIX lint*	295	144
PC-LINT..*by Gimpel, subset of UNIX Lint*	139	125
COBOL		
Micro Focus COBOL/2	900	795
Micro Focus Toolset	900	795
Micro Focus Personal COBOL	149	134
Microsoft COBOL inc. COBOL Tools	700	Call
for XENIX	995	Call
RM/COBOL...*by Ryan McFarland*	950	675
RM/COBOL 85...*ANSI 85*	1250	995
COBOL Support		
Btrieve Softcraft's File Manager	245	169
GSS CGI....*Device independent graphics*	495	375
Halo.......*from Media Cybernetics*	300	219
RM/Screens..*Screen generator*	395	335
RM/Net+ 5...*R/M COBOL networking*	300	249
DBASE & RELATED PRODUCTS		
Applications Plus....*Fox & Geller*	499	279
Brief & dBrief..*Editor/Macro lang for DBase*	275	Call
Clipper.....*Nantucket's DBase Compiler*	695	Call
DATA-p..........*Wallsoft*	60	50
dBC III Plus supports multiuser commands	750	499
with Source code	1500	995
dBC ISAM...*accesses dBase files*	250	169
with Source code	500	349
dBX Translator......*dBASE to C translator*	550	419
with Library Source	950	725
dFlow.......*Wallsoft*	149	124
Documenter.......*Wallsoft*	295	244
FoxBase+.....*Fox Software*	395	Call
Multiuser version	595	Call
QuickCode Plus........*Fox & Geller*	295	170
QuickEntry...........*Fox & Geller*	99	59
QuickReport..........*Fox & Geller*	295	170
Source Print...*Aldeberan's beautifer*	99	75
Tree Diagrammer..*Also from Aldeberan*	77	67
UI Programmer....*Wallsoft*	295	244
FORTRAN Compilers & Utilities		
GSS Graphics Development Toolkit....*CGI*	495	375
GSS GKS...*Kernel Sys,ANSI Level 2b*	495	375
Halo.......*from Media Cybernetics*	300	219
Microsoft Fortran..*Ver 4.0,inc. Codeview*	450	Call
for XENIX	695	Call
R/M Fortran...*ANSI 77 by Ryan McFarland*	595	399
for XENIX	750	599
Spindrift Library....*By Spindrift Labs*	149	129
Source Print	99	75
Tree Diagrammer	77	67
PROLOG		
APT..*PROLOG Tutor*	65	50
Arity PROLOG Compiler & Interpreter	650	569
Arity PROLOG Interpreter	295	229
Arity Standard PROLOG	95	77
PROLOG-86 Plus.........*Solution Systems*	250	199
Turbo PROLOG...*Borland Intl.*	100	63
Turbo PROLOG Toolbox...*Borland*	100	64
TEXT EDITORS		
Brief......*from Solution Systems*	195	Call
dBrief..*Macro lang for Brief & DBase*	95	Call
Brief & dBrief Combo	275	Call
Condor Editor....*Condor Corp*	130	69
Epsilon.....*Lugaru*	195	149

	LIST	US
KEDIT.....*Mansfield..identical to XEDIT*	125	99
Pmate.....*Phoenix*	195	98
Vedit Plus.........*Compuview*	185	129
DEBUGGERS		
Advanced Trace 86.....*Morgan*	175	119
C-Sprite...*Source debugger for Lattice C*	175	139
Periscope I...*Board, Switch, Software*	345	289
Periscope II..*Breakout Switch & Software*	175	139
Periscope II.....*Software only*	145	105
Periscope III....*8 Mhz*	995	795
Periscope III...*10 Mhz*	1095	875
Pfix 86 Plus....*Phoenix symbolic debugger*	395	194
LOGITECH		
MODULA-2 Compiler Package	99	79
MODULA-2 Development Pkg	249	199
MODULA-2 Toolkit	169	139
MODULA-2 ROM Package	299	239
MODULA-2 Window Package	49	39
PHOENIX		
C/PAC........PForce & Pre-C	495	299
Pasm 86......*Macro ASSEMBLER*	195	98
Pdisk......*Disk Management Utility*	145	89
PFantasy.....*six-pack take-away*	995	549
PFinish......*Profiler*	395	194
PFix 86 Plus.Symbolic Debugger	395	194
PForCe.......*Utility library*	395	194
PforCe...*Pforce for C*	395	194
PLink 86+...*sophisticated overlay linker*	495	259
PMaker.......*make utility*	125	69
Pmate........*Text Editor*	195	98
Pre-C........*Super-set of UNIX Lint*	295	144
Ptel.........*Binary Transfer Program*	195	39
POLYTRON		
PolyBoost....*Software accelerator*	80	64
PolyDesk III..*3rd Generation Desktop org*	99	73
PolyLibrarian...*Library Manager*	99	73
PolyLibrarian II	149	109
PolyMake......*Complete MAKE Utility*	149	109
PolyShell.....*UNIX-like Command Shell*	149	109
PolyXREF2...*Cross Reference Util all lang*	219	169
PolyXREF2...*Single Language support*	129	99
PVCS Corporate...*Source Code Control*	395	309
PVCS Personal...*Personalised ver of above*	149	109
PVCS Network...*Powerful Ver. of PVCS*	Call	Call
RYAN McFARLAND		
RM/COBOL......*ANSI 74 Standard*	950	675
for UNIX or XENIX	1250	999
RM/COBOL 85.....*ANSI 85 Standard*	1250	995
RM/FORTRAN.....*ANSI 77 Standard*	595	399
for UNIX or XENIX	750	599
RM/NET+ 5...*COBOL Networking*	300	249
RM/Screens.....*COBOL 85 Screen generator*	395	335
SOFTCRAFT		
Btrieve Softcraft's File Manager	245	169
Xtrieve...*Query language for Btrieve*	245	220
Report Option for Xtrieve	145	128
Btrieve/N File Management for Networks	595	429
Xtrieve/N.....*Multi-User Query*	595	459
Report Option/N...*Multi-user Rep Opt*	345	269
XQL......*SQL for Btrieve*	795	595

SUPER SPECIAL...50% Discount

Condor Editor... An unbeatable editor with more commands, more features, more "Bells & Whistles", no matter what language you use. Normally **$130.00**, New price **$69.95**.

Limited stock, call for availability.

PRICED TO SAVE, SHIPPED FAST!!!

HALO

Graphics Tools for Software Developers

When choosing your graphics tools for software development, performance is crucial. The best proof of HALO's superior performance is in the number of Independent Software Vendors (ISVs) that develop and distribute HALO-based applications. Over 190 ISVs selected HALO because: HALO out performs the rest; HALO is easier to program and shortens time-to-market; HALO is complete with over 190 functions; 17 programming languages; and HALO supports a wider range of devices—over 110.

Since 1981 HALO has been the industry standard library of graphic subroutines for the PC. HALO has an installed base of 60,000+ end users and hundreds of site licenses.

A single copy of HALO includes all device drivers, your choice of one compiler binding, LearnHALO (an interactive tutorial), free 800 # technical support, and the license to use HALO on one machine. Flexible, practical licensing terms are available for ISVs and "Sites."

	List:	PC Express:
Halo	**$300**	**$219**
with Dr. Halo III	**$440**	**$299**
Halo for Microsoft	**$595**	**$434**

GREENLEAF LIBRARIES

Functions

C source, assembler source, and binary libraries of 225 functions for many compilers. Emphasizes tight functional groupings to minimize loading code which your application may never use. Manual's 250 pages help select functions, as do demos, bulletin board.

Communications

Communicate from within your own C programs! Over 120 functions and demo programs in C and assembler source to set up interrupt-driven asynch communications for up to 16 channels. Up to 9600 baud, ASCII or binary, any parity or word length, 8250 UARTs, Xon/Xoff and Xmodem, WideTrack receive. Goodbye separate communications software. Specify compiler.

Data Windows

Windows, menus and data entry do work together...when you utilize Greenleaf's screen architect. This smooth screen designer offers device independence, logical windows, table driven data entry and economical pricing. Source code is also available.

	List	Ours
Communications	**$185**	**$139**
Functions	**$185**	**$139**
Data W indows	**$225**	**$169**
Complete 3 in 1 Pack	**$595**	**$399**

ARITY'S

New Generation of Software Development Tools

Arity's advanced software technology brings to PC'S state of the art applications which previously required mini computers or special processors. Arity products combine not only the fastest benchmarks but C-Language integration, virtual memory, SQL, a sophisticated DBMS, presentation/window style interfaces, and expert systems to give your applications a competitive edge.

ARITY PROLOG Version 5, COMPILER/INTERPRETER or INTERPRETER

Most powerful Clocksin-Millish superset Prolog available. Is now the fastest, with a naive reverse benchmark of 17320 logical inferences per second on an 8 megahertz Compaq 286. Includes virtual database of up to 2 gigabytes with b-trees, hashing, built in editor and symbolic debugger, compile arithmetic, online help, over 200 predicates for windows, menus, dialog boxes, edit field, and list management. Version 5 supports embedded C expressions, global variables and structures, C data types, and calls to Prolog from C or vice versa. List: **$650**, PC Express: **$569**.

ARITY PROLOG ADVANCED TOOLBOX

Four groups of routines for building advanced applications with Arity Prolog version 5: a screen and report designer; file interchange routines for Ascii, 1-2-3TM, SymphonyTM, and dBASETM; a complete working spreadsheet; and miscellaneous data management routines. List: **$150**, PC Express: **$119**.

ARITY/SQL

A complete implementation of the IBM standard query language for OS/2. Arity/SQL maintains tables, indexes, and views. Queries can be accepted in SQL syntax or translated to SQL from menu selections or from another query language. Using Arity/SQL, tabular data can be extracted from any SQL compatible data base, manipulated in spreadsheets or PC data bases, then exported back. Available as a C callable library (requires a C compiler) or Prolog callable routines (requires Arity Prolog). List: **$295**, PC Express: **$229**.

ARITY/EXPERT

A set of advanced tools for building applications which approximate human decision making. Supports inexact reasoning, knowledge representation by either frame-based taxonomies or rules, interfaces to C and Pascal, 1 gigabyte of virtual memory and up to 20000 rules. Requires Arity Prolog. List: **$295**, PC Express: **$229**.

ARITY COMBINATION PACK

Groups all of the products described above, in a complete integrated development workbench. List: **$1095**, Us: **$979**

GSS GRAPHICS SYSTEM

Leave the Device Driving to GSS

GSSTM has reconfigured two components of its comprehensive graphics tools to conform with the ANSI Computer Graphics Interface (CGI) standard.

At the heart of the system is the Development Toolkit which contains all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras, and more. Drivers house management of vector graphics (plotters) and bitmaps used by raster input devices (scanners) to insulate the application program from concern for device idiosyncracy. No one else has implemented CGI that way. It means your programming remains generic; just switch drivers and the same program will drive a different device.

GSS KernelTM conforms to level 2b of ANSI's Graphical Kernel System (GKS) and contains all its needed drivers and language bindings. Kernel has macro level tools to draw and color an object, store the sequential instructions, and recreate the object on its own, as well as segment it, transform it, etc. So powerful, a single command may represent several score lower level statements.

Kernal has the tools for graph and chart generation and their captioning: hand it apples and oranges, say "pie", and it bakes the numbers into a digestible display for screen or plotters.

Kernel can convert the images it creates to ANSI Computer Graphics Metafiles (CGMs), a tokenized standard for storing every form of graphic image as data. The Metafile Interpreter reads the contents of a CGM and interprets it with full CGI capability for recreation on various devices.

Quality software? IBM thinks so. They sell the GSS series under their own label.

Unit royalties and annual fees have been instituted for redistribution. Needs 256k.

Ask for:	List	PC Express:
CGI Dvlpmt Toolkit	**$495**	**$375**
Kernel System	**$495**	**$375**
Kernal for IBM RT	**$795**	**$645**
Metafile Interpreter	**$295**	**$235**

BLAISE C TOOLS PLUS/5.0

C TOOLS PLUS/5.0 from Blaise Computing Inc. helps you to quickly build professional applications using the full power of Microsoft C 5.0 and QuickC. Now you can concentrate on program creativity by having full control over DOS, menus, interrupt service routines, memory resident programs, fast direct video access; windows; printer and keyboard control, and more!

Blaise Computing's attention to detail, like the use of full function prototyping, cleanly organized header files, and a comprehensive, fully-indexed manual, makes C TOOLS PLUS/5.0 the choice for experienced developers as well as newcomers to C.

C TOOLS PLUS/5.0 prebuilt libraries are ready to use with either QuickC or the Microsoft C 5.0 command line environment. Complete documented source code is included so that you can study and adapt it to your specific needs.

	List:	PC Express:
C TOOLS PLUS/5.0	**$129**	**$ 99**
Turbo C TOOLS	**129**	**99**
C ASYNCH MANAGER	**175**	**135**
Turbo POWER TOOLS	**99**	**75**
Turbo ASYNCH PLUS	**99**	**75**

ESSENTIAL C UTILITY LIBRARY

400 Functions, 30c Each

You've probably seen the speed and power of Essential's C function library without knowing it. Software greats have been using it for some time to give today's top products pizazz and panache.

Now grown to 400 functions Essential produces pop-up menus, saves and restores screens and windows to disk or memory in as little as 1/10th second, and claims the fastest video output available. Library has 50 business graphics functions, 40 string handlers, 28 functions for printers, 18 for mice, 11 for time and date, DOS interface functions for disk error trapping, directory and file creation and management, lots more. Everything in source, including sample programs. We have versions with pre-built libraries for the well-known C compilers, and a source code librarian is supplied for rolling your own.

	List:	PC Express:
C Utility Library	**$185**	**$119**
Essential Graphics	**$250**	**$183**
Essential Communications	**$185**	**$125**
with Breakout Debugger	**$250**	**$189**

SUPER SPECIAL...50% DISCOUNT

Condor Editor....An unbeatable editor with more commands, more features, more "Bells & Whistles", no matter what language you use. Normally $130.00, New price $69.95.

Limited stock, call for availability.

MICROSOFT C

The Keeper of the Technology Takes Over

It bundles a source debugger and a "make", and sports a "huge" memory model permitting single data objects larger than 64k, but what's really impressive about Microsoft C are the benchmarks reported in Dr. Dobb's. Microsoft runs away from a field of 17 winning 11 of 27 benchmarks.

The CodeViewTM debugger uses windows to show everything on one screen: source alongside disassembled object, variables, stack and registers. Drop down windows obviate learning of commands. "A source-level debugger that puts the rest to shame" said Dobb's.

Microsoft C has five memory models for code and data, plus non-library support for another thirteen, and boasts alternate math packages for speed versus accuracy, with or without 8087/80287 chips.

Both linker and library manager are part of the package, as is the "make", which knows how to rebuild any size project by compiling only elements which have changed.

It is reportedly used by Lotus, Ashton-Tate and, fittingly, Microsoft itself to develop Windows. Dobb's calls it "the best MS-DOS C development environment value today [for] virtually any kind of program conceivable." 320k suggested.

List: **$450**, PC Express: **$295**.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Express, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.
Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $12, each add'l $6. UPS Next Day Air and Federal Express shipment costs based on weight. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders or Literature, Call Us Toll Free at...

PC EXPRESS™

A Division of PC BRAND, 150 Fifth Ave., New York, NY 10011

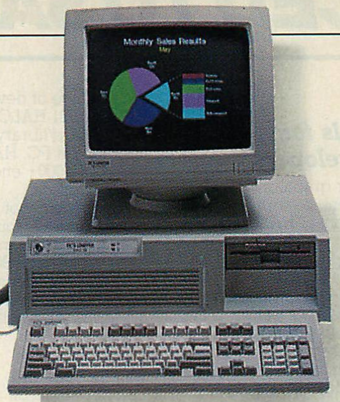
1-800-321-7733

PHONE: (212) 242-3600; FAX: (212) 627-1171;
TELEX: 667962 (SOFT COMM NYK)

CIRCLE NO. 233 ON READER SERVICE CARD

TECH RELEASES

The latest in hardware, software, and technology for systems developers and integrators



Dell Computer Corporation's PC's Limited System 200

SYSTEMS

Dell Computer Corporation has announced that it is replacing the PC's Limited 386[16] with the **PC's Limited System 300**, an 80386-based microcomputer operating at 16 MHz. Standard features include 1MB of static RAM expandable to 6MB of SRAM and 10MB of DRAM; a 1.2MB 5.25-inch or 1.44MB 3.5-inch diskette drive; one parallel and two serial ports; a socket for an 80287; eight expansion slots; a 101-key keyboard; and a 12-month, on-site Honeywell Bull service contract. The System 300 also includes the **Dell System Analyzer**, an advanced set of diagnostic tests that check the memory, video card and monitor, keyboard, DMA controller, parallel and serial ports, modem, diskette drive, and hard disk. Prices range from $4,299 for a 40MB hard disk, monochrome system to $6,699 for a 150MB hard-disk, VGA Color Plus system.

Replacing the PC's Limited 286[8] and 286[12], the **PC's Limited System 200** features a 12.5-MHz clock speed; 640KB of RAM expandable to 4.6MB on the system board and 16MB total memory; choice of 1.2MB 5.25-inch or 1.44MB 3.25-inch diskette drives; one parallel and two serial ports; six expansion slots; an enhanced 101-key keyboard; and a 12-month, on-site Honeywell Bull service contract. The Dell System Analyzer is also standard on System 200 models. The System 200 can be configured with a 20MB, 40MB (40ms), 40MB (28ms), 70MB, or 150MB ESDI hard-disk drive. A choice of available monitors includes monochrome, EGA, VGA monochrome, or VGA Color Plus. Prices range from $1,799 for a monochrome system with a 20MB hard-disk drive to $4,699 for a VGA Color Plus system with a 150MB ESDI hard-disk drive. Dell also announced an enhanced version of Microsoft's **MS-DOS 3.3**, which includes a set of 15 utility programs exclusively for Dell Computer customers. $119.95.

Dell Computer Corporation, 9505 Arboretum Blvd., Austin, TX 78759-7299; 512/338-4400

CIRCLE 301 ON READER SERVICE CARD

CONNECTIONS

An Ethernet adapter for Apple Computer Inc.'s Macintosh II that allows the computer to operate on a 10-Mbps Ethernet network has been announced by **3Com Corporation.** The **EtherLink/NB** is designed for use with 3Com's **3+ for Macintosh network operating system**, and is compatible with Apple's AppleShare network operating system. EtherLink/NB employs the Macintosh II's NuBus expansion slot and uses a 16KB packet bus. The

3Com's 3+Mail 1.3 for PCs and 3+Mail for Macintosh

NuBus interface provides 32-bit data transfers. EtherLink/NB, $595; enhanced 3+ for Macintosh operating system, $495; upgrades, $400.

3Com also announced two 3+Mail products that allow integrated electronic mail service between Macintosh and PC environments over Ethernet, LocalTalk (AppleTalk), and Token-Ring networks for local and remote locations. Together, **3+Mail for Macintosh** and **3+Mail 1.3** enable users to transmit electronic mail transparently among dissimilar workstations and across dissimilar network topologies. Transmission is also transparent to interconnected and remote networks. Users can create multiple folders to store mail, transfer attachments, and receive automatic notification of new mail. 3+Mail for Macintosh and 3+Mail 1.3 for up to five users, $595. Service to unlimited users is enabled by two copies of either product or one of each, $1,190.

3Com Corporation, 3165 Kifer Road, Santa Clara, CA 95052-8145; 800/638-3266; 408/562-6400

CIRCLE 302 ON READER SERVICE CARD

Apple Computer Inc. has introduced **MacAPPC**, its software implementation of IBM's logical unit 6.2 and physical unit 2.1 protocols, which are key parts of IBM's Systems Network Architecture. With MacAPPC, Apple is including HyperCard APPC, a HyperCard stack development tool that lets programmers create applications that can include graphics and sound as well as text, and can navigate through large amounts of information. Site license, $2,500.

Apple also introduced **MacWorkStation**, a software toolkit that lets developers, resellers, systems integrators, and MIS organizations create a Macintosh interface for programs operating on host computers. Price for internal use, $2,500; for commercial resale, $5,000.

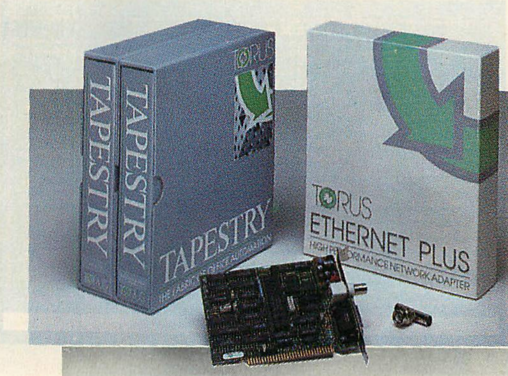
AppleTalk for VMS 2.0, a software system that runs as a process under Digital Equipment Corporation's VAX VMS operating system, was also introduced. Site license, $5,000.

Apple Computer Inc., 20525 Mariani Avenue, Cupertino, CA 95014; 408/996-1010

CIRCLE 303 ON READER SERVICE CARD

The **TCP/IP Workstation** for Novell's NetWare 2.0a and 2.1 has been announced by **MICOM-Interlan Inc.** The TCP/IP workstation is a hardware/

Esprit's LAN TERM processing terminal



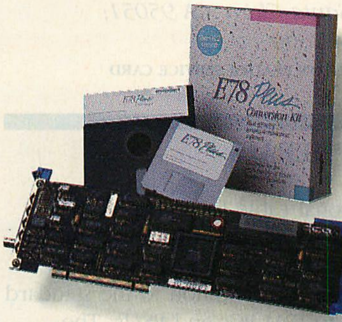
Torus Systems's Tapestry/8 and Ethernet Plus adapter

software solution for users who need TCP services primarily and NetWare services secondarily. Users can transfer files between any TCP/IP host and NetWare file server or can exchange files between their own PC and any other computer system that supports the TCP/IP protocol. The applications provided include FTP file transfer, TELNET terminal emulations, SMTP electronic mail, and a set of TCP utility applications. VT100 and IBM 3270 terminal emulation is included to allow host terminal session login. $795.
MICOM-Interlan Inc. (a subsidiary of MICOM Systems Inc.), 155 Swanson Road, Boxborough, MA 01719; 800/526-8255; 617/263-9929
CIRCLE 304 ON READER SERVICE CARD

For users who want to run PC applications on their desktop as well as share resources and files in a LAN, **Esprit Systems Inc.** has announced its **LAN TERM** processing terminal. The LAN TERM includes an 8086-compatible CPU running at 8 MHz or 10 MHz and a built-in Arcnet interface for integration into a Novell NetWare-based network. It also includes an RS-232 port for direct connection to a host, mouse, or other serial device, and a parallel port for access to a printer. The LAN TERM offers 768KB of RAM, allowing users to access up to 640KB of RAM for applications; the remaining 128KB of RAM supports the network software and terminal emulation functions. An Intel 8087-2 coprocessor is optional. $1,095.
Esprit Systems Inc., 100 Marcus Drive, Melville, NY 11747; 516/293-5600
CIRCLE 306 ON READER SERVICE CARD

An eight-user version of the Tapestry network system has been announced by **Torus Systems Inc. Tapestry/8** is LAN software that may operate with Torus's own LAN hardware, the Torus Ethernet Plus adapter, or with any other LAN hardware that provides a NETBIOS. Tapestry provides electronic mail, file sharing, printer sharing, network management, and asynchronous communications with an icon user interface that promotes ease of use. The Torus Ethernet Plus adapter, which transfers data at 10Mbps, is available for $395. Tapestry/8, $795.
Torus Systems Inc., 240 B Twin Dolphin Drive, Redwood City, CA 94065; 800/872-5335; 415/594-9336
CIRCLE 307 ON READER SERVICE CARD

Digital Communications Associates Inc. (DCA) has announced that it has begun to ship its **E78 Plus for OS/2** software, which is designed to provide 3270 terminal emulation for IRMA and Forte PJ users under the new OS/2 operating system. E78 Plus for OS/2, when used in conjunction with DCA's



E78 Plus for OS/2 software from DCA

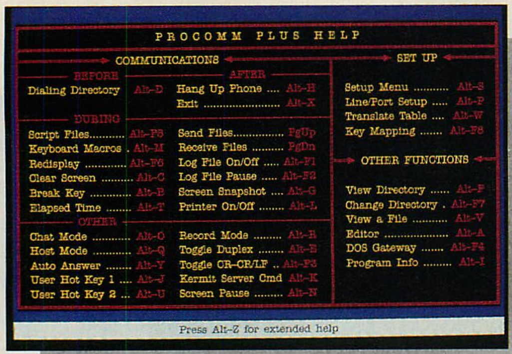
3270 boards, provides IBM PC or PS/2 computer users with character-based CUT-mode 3278/79 terminal emulation running in the OS/2 protected mode. The product includes DCA's IRMAlink FT/TSO and IRMAlink FT/CMS high-speed file-transfer programs, which provide background file transfer under OS/2. $395; for proven users of a DCA emulator, $75.

Windowlink for IRMA, a DCA software product that allows IRMA to run as a true Microsoft Windows application, is also available. The product provides the user with a graphical user interface that parallels that of all Windows applications. $195.

DCA also announced a Micro Channel version of **IRMA 3279 Graphics**, which allows IBM PS/2 models 50, 60, or 80 to emulate an IBM 3279 S3G color terminal, thus enabling the user to create mainframe color graphics in any IBM 3270 processing environment. IRMA 3279 Graphics consists of an IRMA 2 board with an attached daughterboard that provides the graphics terminal emulation. Bundled with the product is DCA's IRMAlink FT/TSO and FT/CMS for high-speed file transfer. $1,995.
Digital Communications Associates Inc., 1000 Alderman Drive, Alpharetta, GA 30201-4199; 800/241-4762; 404/442-4000
CIRCLE 308 ON READER SERVICE CARD

Novell Inc. has introduced **NetWare LU6.2**, a LAN-to-host connectivity software product that provides NetWare LANs with a link to IBM SNA hosts. NetWare LU6.2 will support a wide range of applications, from file transfer to program-to-program communications, and is fully compatible with IBM's APPC/PC. The product is included without charge with all NetWare Gateway products; NetWare LU6.2 upgrades for existing gateways, $100.

Also announced by Novell is the **NetWare Token-Ring Gateway**, a direct Token-Ring connection via an IBM 3725 front-end processor or 3174 communications controller, and the **NetWare Token-Ring Multi**, a multiple-session micro-to-mainframe software package that uses the IBM Token-Ring LAN to provide a direct connection to IBM SNA hosts. NetWare Token-Ring Gateway, $550; NetWare Token-Ring Multi, $395.
Novell Inc., 122 E. 1700 S, Provo, UT 84601; 801/379-5900; 800/453-1267
CIRCLE 309 ON READER SERVICE CARD



Datastorm's PROCOMM PLUS help screen

Princeton Graphic System's Ultrasync monitor

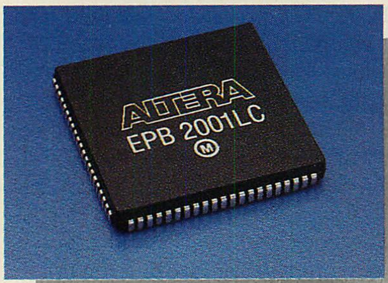
A full-featured stand-alone communications software product, **PROCOMM PLUS**, has been announced by **Datastorm Technologies Inc.** PROCOMM PLUS features 12 error-correcting file-transfer protocols, with two designed specifically to accommodate high-speed error-correcting modems; 16 asynchronous terminal emulations; a fully automated dialing directory; and ASPECT, a powerful script-command language. The program also includes context-sensitive help, keyboard remapping, record (learn) mode, and host mode for remote access. $75.

Datastorm Technologies Inc., 1621 Towne Drive, Suite G, Columbia, MO 65202; 314/474-8461

CIRCLE 310 ON READER SERVICE CARD

TECHNOLOGY

A user-configurable interface chip for use in the design of Micro Channel boards has been announced by **Altera Corporation**. The **EPB2001** replaces the 14 or more TTL and PLD integrated circuits that are needed to build the Micro Channel interface on PS/2 add-in



Altera Corporation's EPB2001 interface chip

boards. The on-chip programmable features of the device will allow it to be adapted to a wide range of different add-in board designs such as disk controller, laser printer controller, graphics controllers, and modems. Other features of the EPB2001 include 2 bytes of EPROM for board ID, four required programmable option select (POS) registers, board bus-control logic, eight programmable chip-select blocks, and 16 POS I/O lines to replace jumpers on the add-on board.

The **EPB2002** DMA arbitration chip, also introduced by Altera, works with the EPB2001 interface chip to integrate the additional logic necessary to add DMA capability. The DMA arbitration chip allows an add-in board to request use of the Micro Channel bus for high-speed transfers to or from the PS/2 computer memory without disturbing the PS/2's microprocessor. The total area for the two Altera devices is approximately 2 square inches. Price in volumes of 10K and above: EPB2001, less than $12; EPB2002, less than $5.

Altera Corporation, 3525 Monroe Street, Santa Clara, CA 95051; 408/984-2800

CIRCLE 317 ON READER SERVICE CARD

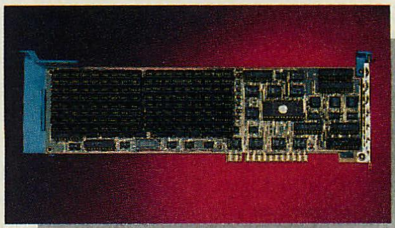
PERIPHERALS

An optional cable for **Princeton Graphics Systems' Ultrasync monitor** allows connection to the Apple Macintosh II in addition to the standard cabling to IBM PCs and PS/2s. The Ultrasync monitor is compatible with MDA, Hercules, CGA, EGA, PGC, VGA and MCGA, and features a .28-mm dot pitch, 800-by-600 pixel resolution, and automatic screen adjustment in all modes. Ultrasync monitor, $849.00; optional cable, $19.95.

Princeton Graphic Systems, 601 Ewing Street, Bldg. A, Princeton, NJ 08540; 800/221-1490; 609/683-1660

CIRCLE 312 ON READER SERVICE CARD

Profit Systems Inc. introduced the **ELITE 16/2**, a memory adapter for the IBM PS/2 Micro Channel that delivers 16MB of OS/2 extended memory or LIM EMS expanded memory for IBM PS/2 Models 50 and 60. ELITE 16/2 breaks the current 7MB barrier on the IBM Model 50. It provides a complete T-RAM implementation, including paging the entire 16MB address area in



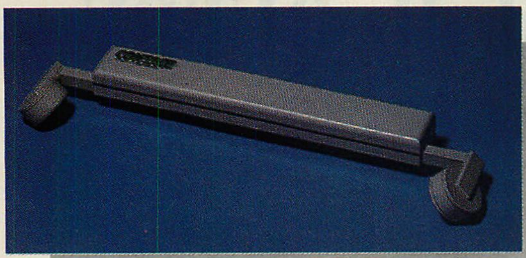
Profit System's Elite 16/2 memory adapter

16KB increments and automatic removal of bad RAM in 16KB increments. The ELITE 16/2 is also compatible with earlier releases of the expanded memory specification. 512KB ELITE 16/2, $595; 512KB and 2MB memory kits, $200 and $1,000, respectively.

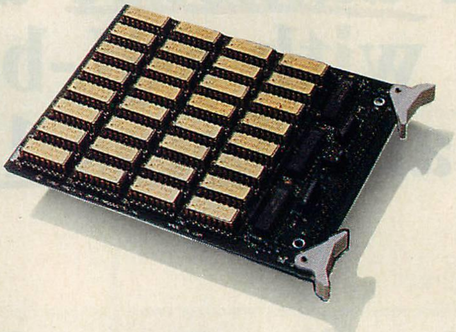
Profit Systems Inc., 30150 Telegraph Road, Birmingham, MI 48010; 313/647-5010

CIRCLE 313 ON READER SERVICE CARD

The **VGA Display Adapter** and the **VGA/AD**, two new graphics boards based on the Tseng Labs VGA chip providing both VGA and EGA hardware compatibility, have been announced by **Tecmar Inc.** Both models support all 17 VGA display modes and IBM EGA, CGA, MDA, and Hercules monochrome video standards. Tecmar's VGA Display Adapters can drive the new IBM analog monitors and fixed- and variable-frequency digital monitors. An exclusive **Turbo BIOS** feature is a RAM-resident utility requiring less than 1KB of memory that improves the speed and performance by performing all BIOS functions in RAM. The Tecmar VGA line will work with the same software drivers distributed with Tecmar's EGA product line, supporting resolutions of up to



PointScreen ultrasonic screen add-on from Contaq Technologies



1-2-4 memory board from Pacific Data Products

800-by-600 pixels. The VGA/AD supports the IBM 8514 display mode of 16 colors with 1,024-by-768 pixel resolution and includes 8514 mode software drivers for AutoCAD, Microsoft Windows, GEM and other software. The VGA/AD also features an optional VGA video overlay board for interactive applications. VGA Display Adapter, $395; VGA/AD, $545.

Tecmar also announced expansion of its line of Micro Channel memory/multifunction boards for the IBM PS/2 family with two new MicroRAM boards. The **MicroRAM AD** offers a maximum of 8MB of OS/2 addressable memory and comes standard with serial and parallel I/O capabilities. The MicroRAM AD also can be configured as an IBM FEFEH board for use in Models 50 and 60. The **MicroRAM 50/60** offers 2MB of memory expansion and provides full support for IBM FEFEH addresses. 2MB MicroRAM AD, $1,145; plug-in I/O module, $200; 2MB MicroRAM 50/60, $995.
Tecmar Inc., 6225 Cochran Road, Solon, OH 44139-3377; 216/349-0600

CIRCLE 314 ON READER SERVICE CARD

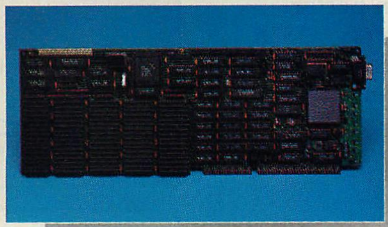
An add-on "touchless" ultrasonic touch screen has been unveiled by **Contaq Technologies Corporation.** The **PointScreen** projects an ultrasonic field in front of the screen; it detects a pointed finger within an inch of the screen and commands the computer to perform the displayed task to which the finger is pointed. PointScreen can be installed on any monitor or terminal (or incorporated in the assembly), and connects simply to any computer through the serial port. $695.
Contaq Technologies Corporation, 15 Main Street, Bristol, VT 05443; 802/453-3332

CIRCLE 315 ON READER SERVICE CARD

A low-cost, upgradable add-in memory board for the Hewlett-Packard Series II Laserjet has been announced by **Pacific Data Products Inc.** The **1-2-4** board is fully compatible with the Series II memory expansion slot. It comes without memory in its standard configuration and uses a unique socket arrangement to accept either 256 kilobit or 1 megabit memory chips. The board can be configured with either 1MB, 2MB, or 4MB and may be upgraded at any time. $295.
Pacific Data Products Inc., 8525 Arjons Drive, Suite M, San Diego, CA 92126; 619/549-0922

CIRCLE 316 ON READER SERVICE CARD

National Design Inc. has introduced its **Genesis 1024** and **Genesis 1280** intelligent PC color graphics controllers, which are bundled with the NOVA*CGI, an implementation of the Computer Graphics Interface (CGI)



National Design's Genesis color graphics controller

developed by Nova Graphics International Corp. Developers using CGI graphics commands can take advantage of the Genesis controller's performance, color, and resolution capabilities, and can perform graphic routines 20 to 40 times faster than on a PC using an EGA card. The Genesis 1024 offers resolutions from 640-by-480 pixels to 1,024-by-768 pixels in 16 colors, while the Genesis 1280 offers resolutions from 640-by-480 pixels to 1,280-by-1,024 pixels in 256 colors. Both use Texas Instruments Inc.'s 34010 graphics system processor operating at 40 MHz or 50 MHz and provide a maximum of 32MB of expandable memory. Genesis 1024, $1,700; Genesis 1280, $2,995 to $3,995, depending on memory configuration; optional EGA emulation board for Genesis 1280, $425.
National Design Inc., 9171 Capital of Texas Highway, North Houston Bldg., Suite 230, Austin, TX 78759; 512/343-5055

CIRCLE 311 ON READER SERVICE CARD

SOFTWARE DEVELOPMENT

Compaq Computer Corporation has announced the delivery of **Microsoft OS/2 Standard Version 1.0** for use with all 80286- and 80386-based Compaq personal computers. MS OS/2 is a single-user, multitasking operating system that allows users to address a maximum of 16MB of RAM. It features a DOS mode for running current MS-DOS-based applications and is fully compatible with applications being developed for IBM OS/2 Standard Edition 1.0. MS OS/2 also provides support for specific features of Compaq machines, such as tape backups. $325.

Compaq also announced a **1.44MB 3.5-inch diskette drive option** for the Compaq Portable 386, Portable III, and Portable II. $245
Compaq Computer Corporation, 20555 FM 149, Houston, TX 77070; 713/370-0670

CIRCLE 319 ON READER SERVICE CARD

A complete Prolog programming environment, **Arity/Prolog 5.0**, has been announced by **Arity Corporation.** The enhanced version includes an embedded C compiler that facilitates the use of C and Pascal languages in conjunction with Prolog. A set of screen-design predicates has been added for creating windows, pull-down menus, and dialog boxes. Other features include arithmetic computation, database management and indexing, and DOS interaction. Arity/Prolog Compiler and Interpreter,

PVCS

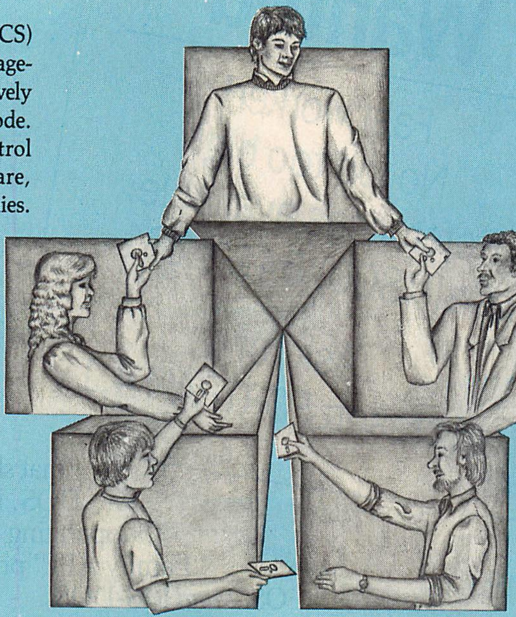
MS-DOS · COMMON INTERFACE & FILE FORMAT · VAX/VMS

The Number One Source Code Control System.

The POLYTRON Version Control System (PVCS) simplifies and automates Configuration Management so programmers and managers can effectively control the revisions and versions of source code. PVCS is the most widely used change control product and is used by the leading software, aerospace, manufacturing and service companies.

"In terms of features, PVCS provides everything necessary to a large multi-programmer project — more than any other package reviewed. No restrictions are placed in the development environment and all aspects of operation can be customized for specific project needs."

PC Tech Journal
September 1987



Unmatched Flexibility

- Storage & Retrieval of Multiple Revisions of Source Code
- Maintenance of a Complete History of Changes
- Control of Separate Lines of Development (Branching)
- Resolution of Access Conflicts
- Optional Merging of Simultaneous Changes
- Release and Configuration Control
- Project Activity Reports
- Management Reports
- Command or Menu Interface

Project Control

PVCS maintains individual archieves of all project components in your system — source code modules, data files, documentation and even object code libraries. These "source documents" can be written in any language or multiple languages.

Fast Retrieval of Revisions

PVCS uses "reverse delta storage" which saves disk space and speeds retrieval of versions of any file in the project database. A delta is the set of differences between any revision and the previous revision. PVCS can rapidly recreate complete versions of any file whether it is the most recent revision of a module or the original version of the entire project. Differences are automatically detected and stored.

A Practical Necessity for LANs

While important for single-programmer projects, PVCS is absolutely essential for multiple-programmer projects and LAN-based development efforts. In a LAN environment, source code files are simply too easy to change. Because any change to any file can have major ramifications, coordinating and keeping a record of changes is critical. Project leaders can determine, on a module-by-module basis, which programmers can access or modify source files, libraries, object code and other files. The levels of security can be tailored to meet the needs of nearly every project. PVCS works on all major LANs including 3Com, Novell and the IBM Token Ring Network.

"PVCS has helped us maintain nearly 90 programs and utilities. Without it we would not have the quality of our upcoming release of NetWare."

Jonathan Richey
Manager, NetWare Utilities
Novell

Adopt PVCS on Your Existing Projects

You can obtain the benefits for your current project without disrupting development, regardless of how long your project has been under way. You can build PVCS archives from revisions stored in your present files or simply adopt PVCS from the current date.

PolyMake Reads PVCS Logfile Format

PolyMake, the leading Make utility, understands the structure of PVCS logfiles and is able to correctly determine the date and time of any revision. This prevents unnecessary operations that occur when the date and time of the complete project archive itself is used as with other make utilities.

Once you standardize on PVCS, the archives used to track and monitor changes are interchangeable between any PVCS product.

Personal PVCS — Offers most of the power and flexibility of Corporate PVCS, but excludes the features necessary for multiple-programmer projects.

Corporate PVCS — Offers additional features to maintain source code of very large and complex projects that may involve multiple programmers. Includes multi-level branching to effectively maintain code when programs evolve on multiple paths.

Network PVCS — Extends Corporate PVCS for use on Networks. File locking and security levels can be tailored for each project.

PVCS for VAX systems — Requires VMS. Uses the same interface and archive format as MS-DOS version. Supports branching and offers file locking and other security features for multiple-programmer projects.

	MS-DOS*	VMS		
	PC/XT/AT	Micro VAX II	VAX 7xx	VAX 8xxx
Personal PVCS	$149			
Corporate PVCS	$395			
Network PVCS	$995**	$4,950	$9,500	$10,500+
PolyMake	$149			
Network PolyMake	$447**	$1,250	$2,375	$2,500+

**5 Station LAN License. Call for pricing on larger Networks.

TO ORDER:
1-800-547-4000
Dept. PTJ

Oregon & Outside USA call (503) 645-1150.
Send Checks, P.O.s to: POLYTRON Corporation, 1700 NW 167th Place, Beaverton, OR 97006

POLYTRON®

High Quality Software Since 1982

CIRCLE NO. 144 ON READER SERVICE CARD

Employment Wanted

Fast worker!
No job too big!
References available:
Call (800) 541-1261

Ask for
Microsoft FORTRAN 4.1

Our FORTRAN is hungry for work. And it can handle any job you've got. Whether you're writing a math-intensive engineering program on your PC or porting a huge scientific application down from a mainframe, Microsoft® FORTRAN Optimizing Compiler version 4.1 is the FORTRAN for you.

It comes with an impressive resume and a great track record. And with new support for the OS/2 systems along with standard DOS support, it's a real team player that fits in with any organization.

Put your FORTRAN development on the fastest track yet.

Microsoft FORTRAN 4.1 generates the fastest FORTRAN programs on a personal computer. Our leading-edge optimizing technology, including loop optimizations, automatically improves the quality of the code it generates.

The result is object code so streamlined, so compact, and so efficient you may think your program is running on a mainframe.

Take on the biggest assignments with OS/2 systems support.

Support for the OS/2 systems gives you new capabilities. Write huge FORTRAN applications that shatter the 640K barrier of DOS. Make direct calls to the operating system. Even create a single "Family API" program that runs under MS® OS/2 *and* MS-DOS®.

When it comes to porting huge programs to and from mainframe and mini environments, Microsoft FORTRAN 4.1 can tackle projects of any size–even programs as large as one gigabyte!

And it's still GSA-certified Full and error-free ANSI FORTRAN 77, with the largest set of IBM® VS and DEC® VAX® extensions available for personal computers.

So Microsoft FORTRAN 4.1 makes short work of any porting chore. And you're assured of the highest level of reliability you can get in a FORTRAN compiler.

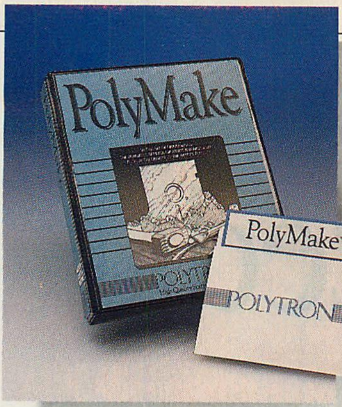
You also get a great set of tools: the new Microsoft Editor, the famous Microsoft CodeView® debugger for both MS OS/2 and DOS, and award-winning documentation.

Give Microsoft FORTRAN 4.1 a long-term contract.

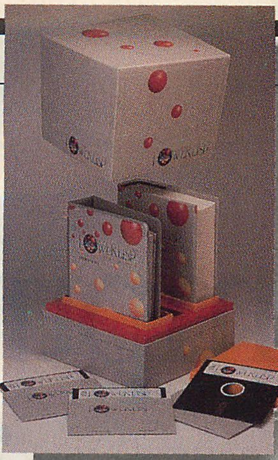
But first check its references for yourself. Just call **(800) 541-1261, Dept. E84,** for the name of your nearest Microsoft dealer. Then put Microsoft FORTRAN 4.1 to work for you.

Microsoft FORTRAN
Optimizing Compiler version 4.1

Microsoft, MS, MS-DOS, and CodeView are registered trademarks of Microsoft Corporation.
IBM is a registered trademark of International Business Machines Corporation. DEC and VAX are registered trademarks of Digital Equipment Corporation.



PolyMake 2.1 Make utility from Polytron Corporation



MicroProducts' PowerLISP 2.0

$650; interpreter alone, $295; Combination Pak (includes Compiler and Interpreter, Arity/Expert Development Package, Arity/SQL Development Package, and the Arity Advanced Toolkit), $1,095.
Arity Corporation, 30 Domino Drive, Concord, MA 01742; 617/371-1243
CIRCLE 324 ON READER SERVICE CARD

An improved DOS Make utility modeled after the UNIX version of MAKE has been announced by **Polytron Corporation. PolyMake 2.1** is 10 times faster and consumes 40 percent less memory than previous versions. Its features include special dependencies for pre- and post-processing, enhanced dependency types and formulations, built-in commands for use in operation lines, conditional constructs, more powerful macros, operation line modifiers, and built-in functions. $149; five-user license, $447; upgrade, $50.
Polytron Corporation, 1700 N.W. 267th Place, Beaverton, OR 97006; 800/547-4000; 503/645-1150
CIRCLE 320 ON READER SERVICE CARD

An enhanced version of **PowerLISP** is set for release by **MicroProducts Inc.** Version 2.0 offers complete support for Common LISP and InterLISP-10, as well as Common LOOPS, which provides a simple method of defining powerful object-oriented programs, with full inheritance among classes of objects. PowerLISP 2.0 features virtual memory support on 80286- and 80386-based microcomputers, an efficient optimizing compiler, and a rich programming environment. 286 version, $2,195; 386 version, $2,995.
MicroProducts Inc., 370 W. Camino Gardens, Boca Raton, FL 33432; 800/553-0777
CIRCLE 323 ON READER SERVICE CARD

A software link that allows microcomputer users to design and test COBOL applications in an integrated workstation environment has been introduced by **Index Technology Corporation** and **Micro Focus Incorporated**. The link converts system designs from Index Technology's Excelerator software into structured skeletal COBOL source code that can be refined using Micro Focus's VS COBOL Workbench programming environment. The link consists of two products: Index Technology's **XL/Interface Micro Focus**, $650; Micro Focus's **Excelerator Interface**, $500.

Excelerator from Micro Focus Incorporated

Index Technology Corporation, One Main Street, Cambridge, MA 02142; 800/227-1412; 617/494-8200
CIRCLE 326 ON READER SERVICE CARD
Micro Focus Incorporated, 2465 E. Bayshore Road, Suite 400, Palo Alto, CA 94303; 800/872-6265; 415/856-4161
CIRCLE 327 ON READER SERVICE CARD

DATA MANAGERS

The **Diamond Release** of **dBXL**, version 1.2, has been announced by **WordTech Systems Inc.** A DBASE work-alike, dBXL offers full compatibility with DBASE; in addition to the extended features of windowing, full on-line help, and automatic memory variable creation, the diamond version offers graphs, desktop publishing, EXPORT commands, multidimensional arrays, user-defined functions and EMS support. $199; dBXL/LAN per server charge (includes Networker Plus), $599.
WordTech Systems Inc., P.O. Box 1747, Orinda, CA 94563; 415/254-0900
CIRCLE 329 ON READER SERVICE CARD

Ashton-Tate has announced **DBASE IV**, the successor to DBASE III PLUS. Designed for computers based on both OS/2 and DOS, DBASE IV has features that include integration of IBM SAA-compatible structured query language (SQL) and a completely redesigned user interface, as well as a new applications generator. It comes with a built-in automatic program compiler that is 10 times faster than DBASE III PLUS. For easy networked multiuser operations, DBASE IV has new record and field-locking capabilities, transaction processing with complete rollback recovery to prevent data loss during a system failure or record lockout, automatic screen refreshing that updates common screens, and eight-layer password protection and data encryption. It is compatible with all previous versions of DBASE III and DBASE III PLUS. $795; upgrade for users of DBASE II, III, or III PLUS, $175 (upgrade for purchasers of DBASE III PLUS between now and the availability date, July 31, 1988, $30).

Ashton-Tate also announced that the **DBASE IV Developer's Edition**, for use by application developers and consultants, will be available on the same date. $1,295.
Ashton-Tate, 20100 Hamilton Avenue, Torrance, CA 90502-1319; 213/329-8000
CIRCLE 330 ON READER SERVICE CARD

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

Computerized Reasoning

With the appropriate knowledge base, user-friendly interface, and inference engine, expert systems imitate human thought to solve difficult problems.

TOM ARCIDIACONO

A physician selects the best chemotherapy drug for shrinking a patient's cancerous tumor. A plant manager monitors industrial operations to detect malfunctions as they happen and shut down operations when needed. A dispatcher routes delivery trucks to optimize mileage and meet overall shipment deadlines. A bank officer analyzes an applicant's credit record to give the green or red light for a loan. A systems expert configures the most cost-effective computer system for a corporation.

Making these decisions can earn (or lose) billions of dollars or even save (or cost) a person's life. Because human experts are scarce in every field, expert systems are stepping in to fill the need.

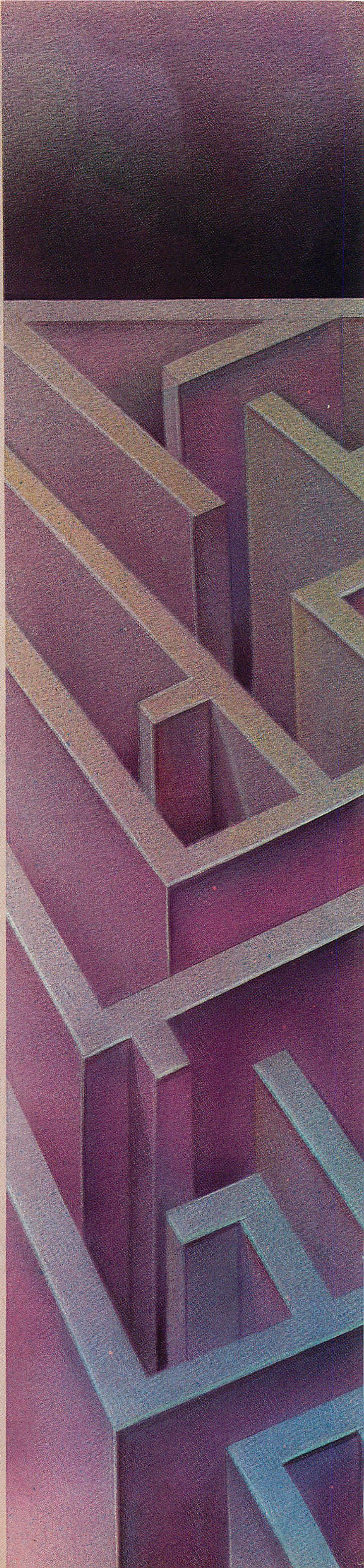
Expert systems, a subfield of artificial intelligence (AI), are computerized problem solvers that analyze massive material on a subject and provide answers. Simulating human reasoning, they find one or several of the best answers hidden in a labyrinth of knowledge. They provide expert advice to nonexperts, assist experts, and act as teaching tools. These systems are best suited for problems requiring large amounts of symbolic, unreliable, and uncertain input together with detailed knowledge of the subject at hand, or *domain* in AI parlance.

Disease diagnosis is a typical problem domain suitable for expert systems. It requires examination of vast amounts of knowledge (from fields such as anatomy, physiology, epidemiology, and pathology) as the domain matures, and the rapid addition of new facts while minimizing impact on existing knowledge. It involves examining a good deal of data in the form of observations and measurements before drawing conclusions.

Diagnosis also demands a high degree of intuition often developed by a physician through experience, and requires analysis of data that are incomplete, unreliable, and largely qualitative (such as "looks very sick" or "sounds delirious") rather than quantitative (body temperature = 102.4 degree Fahrenheit). Unlike expert systems, conventional programs usually do not process qualitative data.

Financial advisement, circuit analysis, flow monitoring, detecting and correcting malfunctions, optimizing operations, and resource scheduling are other suitable domains. Expert systems are characterized by the following:

ILLUSTRATION • ANDY LEVINE



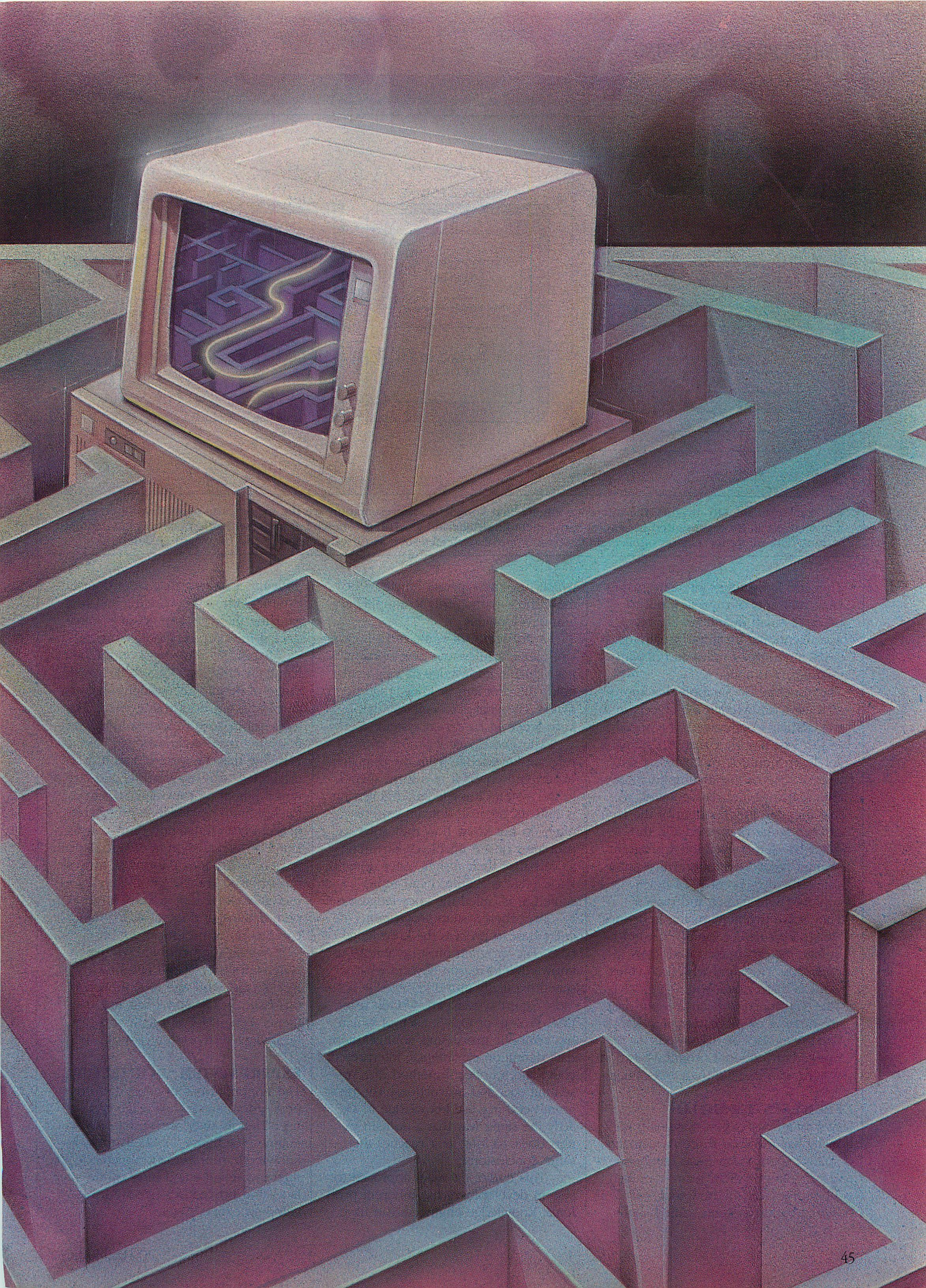
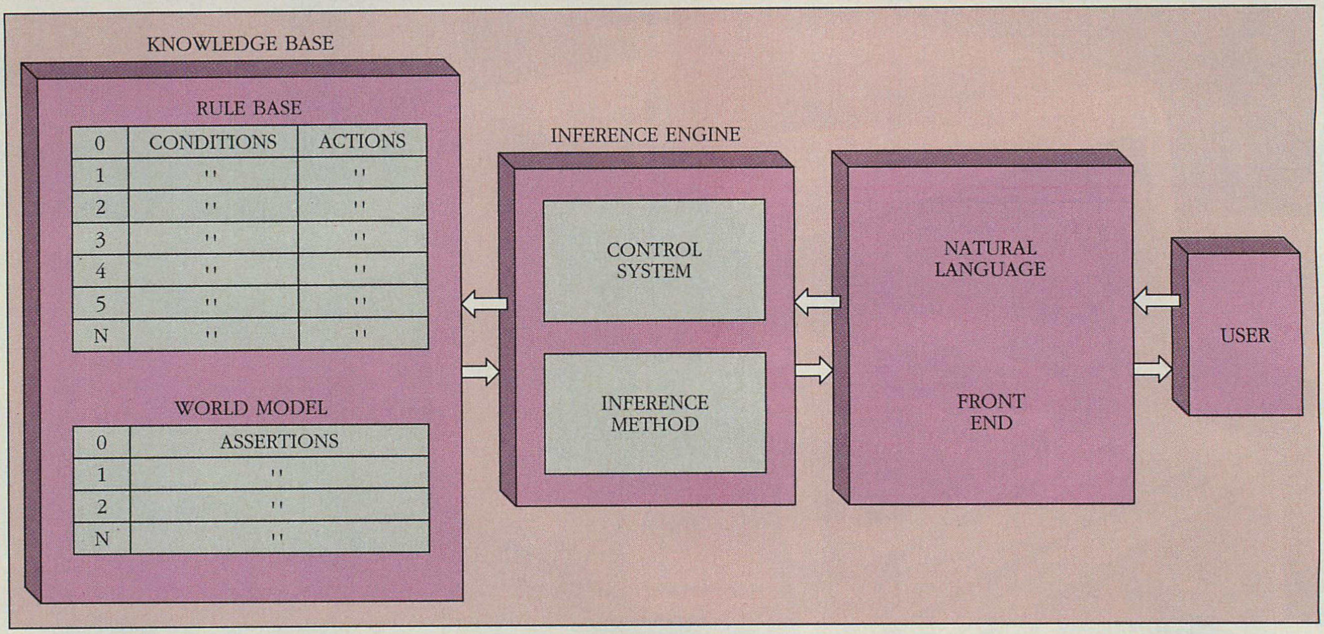


FIGURE 1: ***Expert System Shell Architecture***



The user interface (front end), the inference engine, and knowledge representation are the major parts of shell architecture. Knowledge can be represented in several ways, including rules and assertions (shown), frames, semantic networks, and combinations. The inference engine often includes basic control strategies such as forward chaining and backward chaining.

- They are knowledge intensive, using large amounts of diverse input to produce small quantities of output. A conventional program usually uses small amounts of input and produces volumes of output.
- They operate on symbols that represent facts, knowledge, or objects in a way similar to enumerated type values in Pascal, rather than operating solely on numeric or character data.
- Expert systems use assumptions about data structure, not data content, so that they are independent of the type of knowledge processed. On the other hand, a conventional program using a record definition must know data structure (number of fields) and content (field names and types).
- They solve problems using heuristics, which are rules of thumb that help human experts make rapid leaps in reasoning. How and why such rules work is often unknown.
- They can function even if input data are missing or inaccurate, and knowledge is inexact or disjointed.
- Unlike conventional programs, expert systems can explain their reasoning.

EXPERT SYSTEM SHELLS

Development of expert systems traditionally has involved three participants: domain expert, knowledge engineer, and programmer. This team approach is being replaced by new tools for building expert systems, called shells, which fuse the separate functions of these three participants to simplify system development.

A *domain expert,* the most important element in system development, has valuable knowledge in a particular field (the problem or knowledge domain). The goal is to embody that knowledge in an expert system.

The *knowledge engineer* extracts relevant knowledge from the domain expert and enters it in a form specific to the system. Knowledge engineering is difficult because domain experts often do not know how they solve problems or are unable or unwilling to verbalize their insight. After domain knowledge is acquired and represented, it is called the knowledge base.

An expert system *programmer* decides how to process the knowledge base. The program must be able to use knowledge to reason beyond what is stated explicitly in the knowledge base. Most expert systems are written using LISP, Prolog, or Ops-5, each of which has symbolic processing, sophisticated data definition capabilities, interpreted code, and a uniform method of representing data and instructions. The program need not use knowledge in exactly the same way that a domain expert does, as long as it reaches the same conclusions.

This traditional team approach to system development has inherent problems. Knowledge engineers can have difficulty acquiring data from domain experts without interpreting and reorganizing the data, and possibly introducing misinterpretations.

Expert system shells, on the other hand, are designed for use by a wide range of people, including domain experts who are not programmers. These tools aid in acquiring knowledge, representing it in the knowledge base, and verifying that the system uses it properly. Shells can reduce time and cost of development by providing structures for holding knowledge, a program to process knowledge and draw conclusions, an explanation system to explain reasoning used, extensions to process probabilistic knowledge, methods for incremental development of large expert systems, and methods for rapid completion of prototype systems.

Among the better-known expert system shells are Gold Hill's GoldWorks (see "The Age of GoldWorks," this issue, p. 68) and Texas Instruments' Personal Consultant Plus (to be reviewed in a future issue).

EXPERT SYSTEM ARCHITECTURE

The important components of an expert system are a knowledge base, a user-friendly interface, and an inference engine (see figure 1). Shells provide knowledge representation methods, the inference engine and most of the interfaces and procedures needed to develop an expert system.

The knowledge base contains at least two types of data the expert system uses to solve problems. One describes the problem and comprises information that has been concluded, assumed, or provided during the inference process. This information is contained in an assertion base, or world model. The other type has knowledge about how to use the assertion base to reason about the problem domain.

The user interface allows the user easily to enter facts that describe a problem and to provide information that the system might request during inferencing. The system must communicate clearly with the user. Expert systems often use natural language interfaces to minimize user effort.

The inference engine is the heart of an expert system. The inference engine uses assertions (facts) and problem solving expertise in the knowledge base to infer conclusions. Its design and function depends on types of knowledge structures it must process. Different representation methods require different methods of implementing inferencing.

The most important feature of shell architecture is that the knowledge base is separate from the inference engine. This means that, theoretically, a knowledge base for one domain can be replaced by one from another domain without the need for changes to the inference engine or user interface.

Shells also can include interfaces to other knowledge bases, databases, programs, and languages; automatic knowledge base maintenance; the ability to use combinations of different knowledge structures; programs to acquire knowledge from a domain expert; and code-generation facilities to deliver an expert system onto a computer different from the development machine. Features include menu interfaces, trace and explanation facilities, and on-line help and interfaces to graphics programs, data managers, and spreadsheets. The shells can be used with different knowledge bases without starting from scratch each time.

Data structures representing knowledge in the knowledge base are often termed *knowledge structures*, and their careful selection is crucial to a system's smooth development and ultimate success. Knowledge representations that are natural for a problem domain (or similar to the way a domain expert thinks) are the easiest for domain experts to use. In addition, different knowledge structures allow different kinds of inferences to be made.

FIGURE 2: ***Rule Base***

```
RULE 0:  IF    X is late for work
               X has dirty hands
               X is angry
         THEN  X's car broke down
RULE 1:  IF    X is not at the morning meeting
         THEN  X is late for work
RULE 2:  IF    X's car broke down
         THEN  X has a good excuse for being late
RULE 3:  IF    X has a good excuse for being late
               X is a good employee
         THEN  X will not be dismissed
RULE 5:  IF    X is the boss's son-in-law
         THEN  X is a good employee
RULE 6:  IF    X sold at least one million units
               last year
         THEN  X is a good employee
RULE 7:  IF    X vacationed in St. Croix last year
         THEN  X sold at least one million units
               last year
```

This personnel management rule base represents knowledge in If . . . Then rules used by the inference engine to make decisions about employee X.

Common knowledge structures include rules, frames, semantic networks, logic, procedures, and relational databases.

Rule-based systems depict domain knowledge using If . . . Then rules (production rules). Frame-based systems use structures that are analogous to the Pascal record-type; they can contain more than one type of information, such as character and numeric data, or procedures called under specified conditions. By grouping them into hierarchical structures, frames can then share (inherit) information. This economizes storage and speeds searches.

A semantic network is a more sophisticated structure than either rules or frames. Its simplest form uses nodes in a graph to represent facts or problem states and arcs to represent inferences. The system draws conclusions by finding inference paths between starting nodes and goal nodes. Semantic networks have many variations and often incorporate other structures.

Logic representations are based on predicate calculus, a formal method of representing facts and resulting inferences. Predicate calculus, in fact, is the basis of Prolog.

Procedures, or procedural knowledge representation, are often used in conventional programs. Knowledge of how to solve some small part of the overall problem is contained in a procedure, for example, to calculate averages. Modifying such knowledge is often difficult because interactions between procedures can be complex.

Relational databases are attractive knowledge representation methods because many businesses already store data in this format and because relational databases already contain data relations. How an expert system can use inferencing to find implicit relations in the database is being researched.

Because most popular shells are rule- and frame-based, these knowledge representations are highlighted in this article. The other methods will be considered in the future.

INFERENCING WITH RULES

A rule states that certain conclusions (its *consequent*) will occur if specified conditions (its *antecedent*) are met. The rule's antecedent contains facts or previous conclusions on which the rule's consequent can be concluded. For example:

Rule 1: IF Tom is late for work
Tom's hands are dirty
Tom is angry
THEN Tom's car broke down

Rule 2: IF Location is on the ocean
It is night time
Sky is red
THEN Weather will be good the following morning

When an antecedent is true, the consequent is accepted as true; that is, the rule fires. For example, when Tom has dirty hands, is late for work, and is angry, Rule 1 can fire. The conclusion is that his car broke down.

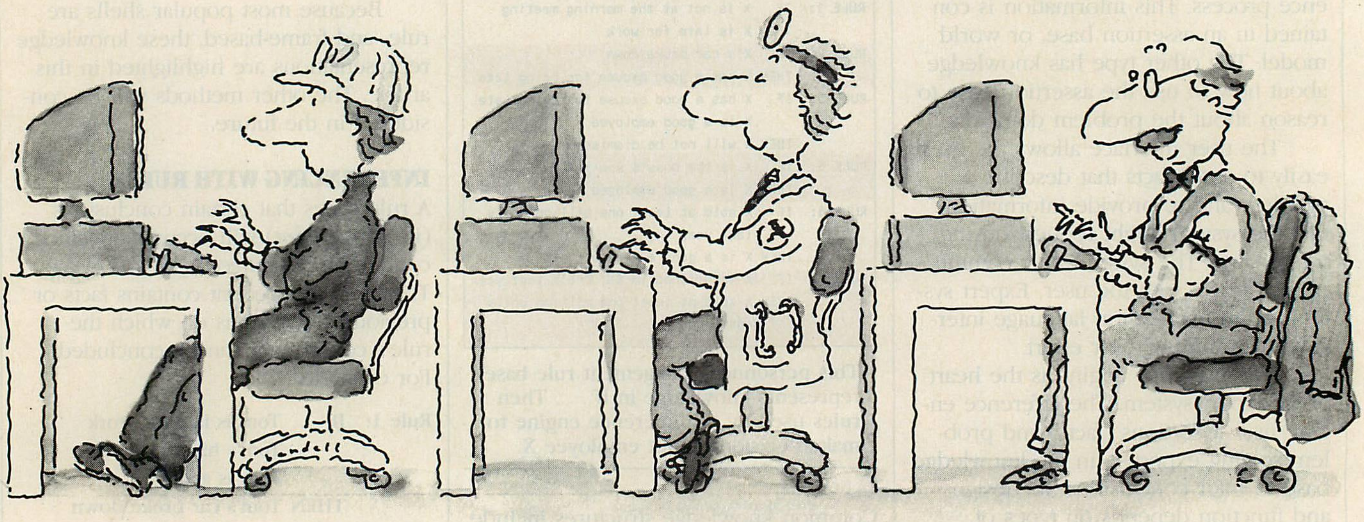
Rule 2 is an example of a heuristic; it is derived from the rule of thumb, "Red sky at night, sailor's delight," and it uses knowledge derived through experience to simplify the search for a solution to the question, "What will tomorrow's weather be like?" Notice that rules contain no apparent information besides the antecedent to justify their conclusions.

In a rule-based system, the inference engine acts on many rules. When one rule fires, its consequent might cause additions to the assertion base that could to satisfy the antecedents of other rules. Consider the following rule using Pascal-like syntax:

```
IF    (coolant-level < optimum-
      coolant-level)
THEN  temperature: = maximum-
      acceptable-temperature
```

The four variables, coolant-level, optimum-coolant-level, temperature, and maximum-acceptable-temperature are facts describing a current problem domain. When a conclusion is reached that affects one of these variables, an assertion is added to the assertion base.

With LEVEL5, you don't have to be an AI expert...



to build an expert system.

LEVEL5 is the one expert system that lets you program a knowledge base fast, even if you're not an AI expert. It uses a clear English-like syntax. And it is powerful enough to handle the most complex applications.

The following is just a sampling of the LEVEL5 applications that are up and running. They're also providing returns as large as 30 or more times the total investment in the project.

Typical Applications

One very large LEVEL5 system uses over 7,000 rules. Written by an auditor, it has become the standard tool for assessing the quality of the loan portfolios of commercial banks.

Another LEVEL5 system helps to determine the manufacturability of engineers' designs. And, LEVEL5 prints out a complete audit trail of its analysis, so engineers know exactly where improvements must be made.

In the computer industry, a system diagnoses problems in large data communication networks. LEVEL5 was chosen because it integrates well with other software.

In fact, fault diagnosis is a major application of LEVEL5, in areas ranging from medical to automotive.

In the mining industry,LEVEL5 has replaced a seven-foot shelf of documents required for compliance with federal mining regulations. Because these regulations change frequently, the ease with which a LEVEL5 knowledge base can be updated is very important.

And similarly in Ireland, LEVEL5 helps architects make sure they are in compliance with the fire code.

In the process industry, LEVEL5 is used for intelligent control of large and complex processes. In marketing, LEVEL5 helps customers select the best paint, pump, or other product for a specific need.

Hardware & Databases

LEVEL5 is the only expert system that runs on the PC and PS/2, the DEC VAX, the Macintosh, and on mainframes*. And only LEVEL5 applications are completely portable across these machines.

On all these platforms, LEVEL5 can directly access existing files on popular database systems.

For more information, or to order PC LEVEL5 at a price of $685 using VISA or MasterCard, call 305-729-9046. Or write Information Builders, Inc. at 503 Fifth Avenue, Indialantic, FL 32903.

*Available third quarter 1988.

LEVEL5
Information Builders, Inc.

CIRCLE NO. 146 ON READER SERVICE CARD

Values can be obtained by querying the user or firing rules. When values for optimum-coolant-level and coolant-level are in the assertion base, the antecedent of the rule is satisfied, the rule fires, and the consequent can be added to the assertion base.

Testing a rule's antecedent can be as simple as matching a symbolic pattern in the rule to a similar pattern in the assertion base. This is called rule pattern matching.

When used with an inference engine, rules in a production system are potentially more powerful than sequences of If . . . Then statements in a Pascal or C program. To draw conclusions from conditional statements in a conventional program, a single consequent must describe each possible conclusion, and a corresponding antecedent must describe every possible condition contributing to that conclusion. Each antecedent-consequent pair must be known in advance by the programmer, and hard-coded into the program.

In contrast, each rule in a rule-based system does not necessarily have to lead to a final conclusion. A rule can instead represent one in a series that must be fired to reach a conclusion. The knowledge engineer need not know rule ordering in advance; the inference engine continually tries to fire rules in the rule base until no more can be fired or until a clearly specified goal assertion or conclusion is made.

Every rule in the rule base must be checked sequentially to see if its conditions can be satisfied by previously made assertions and conclusions. The process continues until every rule is checked, no more rules fire, and no further changes occur to the knowledge base, or until a goal is achieved.

A simple production system strategy is described by the following:

Repeat
Fire-all-rules (Knowledge-base)
Until NOT (changed) or goal-reached;

Consider a simplified knowledge base containing rules about personnel management (see figure 2). Initial assertions in the knowledge base are:

X = Tom
Tom is not at the morning meeting
Tom is angry
Tom vacationed in St. Croix last year
Tom has dirty hands

The inference engine repeatedly scans the personnel rule base looking for rules to fire. Rule 1 is fired first because its antecedent, "Tom is not at the morning meeting," matches an assertion. The system infers that "Tom is late for work" and adds that assertion to the assertion base.

Now that Rule 0 is satisfied, it fires and adds "Tom's car broke down" to the assertion base. Rule 2 fires, adding the assertion that "Tom has good excuse for being late." Rule 7 is satisfied, then Rule 6, and finally Rule 3. The final conclusion is that "Tom will not be dismissed." When the inference process is complete, the assertion base contains the following:

X = Tom
Tom is not at the morning meeting
Tom is angry
Tom vacationed in St. Croix last year
Tom has dirty hands
Tom is late for work
Tom's car broke down
Tom has good excuse for being late
Tom sold at least 1 million units last year
Tom is a good employee
Tom will not be dismissed

In this example, the inference engine skips rules whose antecedents cannot be satisfied by the assertion base. It could search actively for rules that satisfy antecedents, allowing the inference engine to answer questions such as "How can it be proven Tom has a good excuse for being late?"

When no more rules can fire using assertions already available, the inference engine could identify rules that are near firing—for example, those whose antecedents need a single assertion to be satisfied. The inference engine can then use the interface to prompt the expert system user to provide information to verify the assertion.

In this way, an expert system can appear to ask intelligent questions in much the same way that a human expert would. This focuses the user's energy on questions with the greatest chance of solving the problem. This is most appropriate for problems that associate high risk or cost with human observation and measurement.

Rule-based searches can be optimized by flagging already fired rules to prevent them from being reexamined and fired endlessly and by flagging rules whose conclusions cannot add new assertions. Expert systems also might have to choose between conflicting rules. If figure 2 were to have a Rule 8 that says

IF X is late
THEN X will be dismissed

this would conflict with Rule 3, and the system would need to choose between firing Rule 8 or Rule 3. One way the system does this is to allow only one rule to be fired on each pass through the rule base. After identifying each rule that can be fired, it selects one based on *meta-knowledge*, which is knowledge about the nature of the rules themselves. Meta-knowledge could include priorities or other attachments to rules or even separate rules about rules (called *meta-rules*).

Some rules can be specially labeled to indicate that they make final conclusions about a specific problem in the domain. When they fire, the system can relay that conclusion directly to the user. More than one final conclusion can be drawn, given a specific set of assertions about a problem situation. Expert systems are often used this way to identify everything that can be concluded about a situation.

Inference engines commonly allow rules to incorporate some probabilistic information, such as confidence (certainty) factors. Each assertion or conclusion in the database can be associated with a numeric confidence factor; rule antecedents are satisfied only

FIGURE 3: *Sample Frame*

```
NAME:               GYMNOSPERM-FRAME
SPECIALIZATION-OF:  TREE-FRAME
STANDARD SLOTS      VALUES
--------------      ------
TRUNK-DIAMETER
   RANGE:       a real number
   IF-NEEDED:   Multiply HEIGHT by .04
   DEFAULT:     2 feet
 TYPE
   RANGE:    (SPRUCE-FRAME, FIR-FRAME, PINE-FRAME)
   IF-NEEDED:   check needles
                  IF (needle is triangular)
                  THEN PINE-FRAME
                  IF (needle is aromatic)
                  THEN FIR-FRAME
                  IF (needle is flat)
                  THEN SPRUCE-FRAME
   DEFAULT:     SPRUCE-FRAME
 USE
   RANGE:       (swinging, lumber, firewood,
                 target-practice)
   DEFAULT:     firewood
   IF-NEEDED:   IF ACCESSORIES(swing),
                THEN swinging.
                  IF lots of holes in it,
                  THEN target practice
                  IF straight and no knots,
                  THEN lumber
                  OTHERWISE firewood
   IF-ADDED:    IF USE(target-practice)
                  THEN remove ACCESSORIES(swing)
 HEIGHT
   RANGE:       a real number
   IF-ADDED:    Update TRUNK-DIAMETER
 ACCESSORIES    list of [swing,
                clothesline, birdhouse]
```

Frames structure knowledge. Slots, such as Use, describe the object; Facets, such as Range, indicate a slot's value type or procedures.

DOS system running Lotus 1-2-3

SCO XENIX system running SCO Professional

THIS IS AN IBM PS/2 MODEL 80 RUNNING DOS

Under DOS, this PS/2™ is a powerful 80386-based single-tasking, single-user computer that can run thousands of DOS applications. In 16-bit, 8086 mode.

One at a time.

When OS/2™ becomes available, the PS/2 can become a multitasking, single-user computer running in 16-bit, 286 mode that can utilize DOS applications under OS/2.

One at a time.

With DOS or OS/2, the PS/2 will support one user.

	1 user (DOS)	1 user (OS/2)
Cost per system**:	$12,389	$12,594
Cost per user:	$12,389	$12,594

THIS IS AN IBM PS/2 MODEL 80 RUNNING SCO XENIX

Under SCO XENIX,® this PS/2 becomes a powerful 80386-based multitasking, multiuser computer that can run thousands of XENIX applications. In full-tilt, 32-bit, 386 mode.

Many at a time.

And using SCO VP/ix,™* the PS/2 can multitask DOS applications under SCO XENIX.

Many at a time.

With SCO XENIX, the PS/2 will support one user. Or 9 users. Or even 33 users.

And it can do all that today because you can get SCO XENIX for the PS/2—*now!*

	1 user	9 users	33 users
Cost per system**:	$14,559	$19,726	$40,402
Cost per user:	$14,559	$2,192	$1,224

SCO XENIX System V and the SCO XENIX family of software solutions are available for all industry-standard 8086-, 80286-, and 80386-based computers, and the IBM® Personal System/2™ Models 50, 60, and 80.

SCO THE SANTA CRUZ OPERATION

(800) 626-UNIX (626-8649)
(408) 425-7222
FAX: (408) 458-4227
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info

*SCO VP/ix available as separate product.

**Cost comparisons are based on most recently published U.S. domestic suggested list prices. Cost model: Base machine: IBM PS/2 Model 80, 70Mb disk, 1Mb RAM, IBM 8512 color monitor, 1Mb additional IBM RAM, IBM ProPrinter XL,® 1-user DOS system: Base machine, plus DOS 3.3, WordPerfect 4.2, Lotus® 1-2-3,® dBASE III PLUS.® 1-user OS/2 system: 1-user DOS system; substitute OS/2 for DOS. 1-user SCO XENIX system: Base machine, plus SCO XENIX 386 for PS/2, SCO VP/ix, SCO Lyrix® (word processing), SCO FoxBASE+™ (dBASE III PLUS workalike), SCO Professional™ (1-2-3 workalike). 9-user SCO XENIX system: 1-user SCO XENIX system, plus intelligent 8-user multiport card, 8 IBM 3151 ASCII terminals. 33-user SCO XENIX system: 9-user SCO XENIX System, plus 3 more intelligent 8-user multiport cards, 24 more IBM 3151 ASCII terminals, 4 Mb additional RAM, additional 70Mb disk.

CIRCLE NO. 117 ON READER SERVICE CARD

when each component assertion is above a predefined threshold of confidence. When a rule fires, its conclusions can be assigned confidence values based on experience or on confidence values in the antecedent.

The following rule incorporates confidence values (the number in parenthesis is the percentage of certainty for that particular confidence value):

IF X is late for work (100)
X's hands are dirty (50)
X is angry (20)
THEN X's car broke down on the way to work (40)

Rule-based systems have several distinct advantages over other methods of knowledge representation. Rules can be added, modified, or removed—usually without adverse effects on other rules in the rule base. This simplifies the task of knowledge-base construction and maintenance.

Rule-based systems have several limitations, however. They need repeatedly to search the rule base sequentially. Search efficiency decreases linearly with the number of rules. With larger rule bases, this search process becomes unacceptably slow for real-time interaction. Sequential search can be reduced or eliminated by incorporating search methods, such as organizing rules into a discrimination network, where selection of a rule limits subsequent search to a smaller set of rules. Efficiency can approach that of a B-tree under some circumstances, but this requires preprocessing of the rule base. Search time can be reduced also by ordering rules so that the most appropriate rule for a given context will be the first one found and fired by the interpreter. Ordering rules, however, decreases rule independence.

New rules can overlap knowledge represented by existing rules in larger systems. If redundancy becomes acute, system efficiency suffers and knowledge maintenance can become difficult.

The amount of knowledge that can be encoded in a single rule before it becomes unwieldy for humans to read is limited. Partitioning knowledge into multiple rules by increasing the number of rules could slow the system.

Representing procedural knowledge in a production system is difficult (for example, expressing "Voltage = Current * Resistance"). Another limitation is the need to identify sequences of inferences to reach conclusions. Leaps of understanding, to bypass all or part of the inference sequence, are not possible unless coded explicitly as heuristics. Finally, rule-based systems can approach problems from a variety of viewpoints as humans do. They can solve problems from only the perspectives for which they have been coded.

Rule-based systems are used for a wide range of problem domains. A good problem domain for a rule-based system should require seemingly independent assertions and activities. Diagnosing mechanical failures is an example because each step in the diagnosis procedure is distinct; meteorological simulating of weather patterns is a poor example because it involves many interrelated processes.

One of the most well-known examples of a commercially successful expert system using production rules is Digital Equipment Corporation's (DEC) XCON system (developed jointly with Carnegie-Mellon University). Originally called R1, it is a 2,500-to-3,000-rule production system that helps sales and installation personnel configure VAX sites. The program identifies peripherals and other items required to install basic options ordered by customers, and it recommends site placement and layout based on physical and electrical constraints. DEC's use of XCON has reportedly reduced the amount of time, expense, and aggravation needed to configure new systems.

FIGURE 4: *Frame Instance*

```
NAME:            TREE-IN-MY-BACKYARD
INSTANCE-OF:     GYMNOSPERM-FRAME
STANDARD SLOTS   VALUES
-------------    ------
TRUNK-DIAMETER   3 feet
HEIGHT           48 feet
LOCATION         MY-BACKYARD
TYPE             spruce
USE              SWINGING-SCRIPT
NEEDLES          SPRUCE-LEAF-FRAME
```

When actual values are placed into the slots of a frame, an instance of that frame is created. This is a specific object whose general characteristics are given in Gymnosperm-frame.

STRUCTURING WITH FRAMES

Unlike rules, frames provide a method of organizing knowledge into hierarchical structures while retaining a degree of knowledge independence.

A frame is a structure that contains a set of named attributes called *slots*, which together describe a concept, object, or event in much the same way as fields in a record. Slots can be attached to simple assertions, lists, rules, frames, and procedures. Unlike Pascal records, which can contain only declarative information (such as an integer or enumerated value), slots can have procedural information (such as a sequence of instructions to be followed under certain conditions).

Frames can have a variety of slots and facets. The frame in figure 3 contains information about coniferous trees. Gymnosperm-frame has several slots including Trunk-diameter, Type, Use, Height, and Accessories. Many of the slots contain subslots called *facets*, which contain knowledge about the information in slots. Gymnosperm-frame contains commonly used facets, such as Range, If-added, If-needed, and Default.

In Gymnosperm-frame, the Range facet indicates what kind of information can appear in a slot. It is like the type specification for a Pascal record field. The Height slot can contain only a real number, while the Use slot can contain information on different ways the tree can be used. The names might be similar to Pascal enumerated type values, or might represent names of other frames containing more information.

An If-added facet can contain procedural information. It specifies an action to be taken when a value in the slot is added or modified. Such procedural attachments, called *demons*, provide data-driven inferencing by causing procedures to execute in response to adding inferences to the knowledge base. Demons are triggered automatically; the inference engine checks for the existence of an If-added facet when it attaches a new value to a slot.

An If-added facet in the Use slot (see figure 3) contains a set of instructions (in this case, an If . . . Then statement) to be executed whenever the use value is modified. For example, if Use is target-practice, then Swing is removed from the accessory list.

In the Height slot in figure 3, an If-added facet specifies that if the height of the tree is modified, the value attached to the Trunk-diameter slot also should be modified. In the Trunk-diameter slot, the If-needed facet gives a procedure for calculating trunk diameter (multiplying height by .04) if no other value is attached to the slot. Default facets also are available to provide a slot value if no other method is successful or available.

Frames that describe slots and the types of values they have are called generic frames. Frames that represent actual objects are *instances* of a frame. Placing values in slots instantiates (creates an instance of) a frame.

To describe a specific tree, an instance of Gymnosperm-frame, called Tree-in-my-backyard, is created (see fig-

A MODEM THAT IS TWICE AS GOOD AS IT LOOKS.

It says 9,600 bps. But for this modem, transmitting data at 9,600 bps over dial-up lines isn't the half of it.

Its built-in data compression can boost throughput to 19,200 bps. While error-control prevents loss of data.

With no effort on your part—no selecting protocols, no compressing files.

A V-series Smartmodem 9600™ can even take the place of two modems. Because it communicates in either synchronous or asynchronous mode.

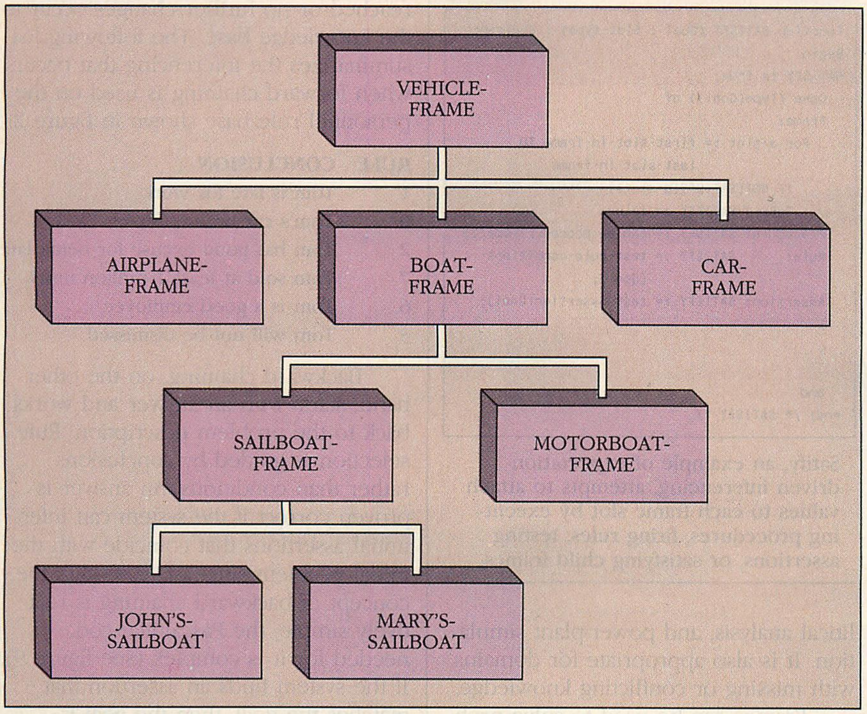
All of which makes our 9600 at least twice as good as it seems. Or, looking at it the other way, more than double your money's worth.

Hayes®

For your nearest Hayes dealer, call **800-635-1225**. Hayes Microcomputer Products, Inc., P.O. Box 105203, Atlanta, GA 30348.

CIRCLE NO. 238 ON READER SERVICE CARD

FIGURE 5: ***Hierarchy of Frames Describing Vehicles***



Frames can be arranged in hierarchies to allow inheritance. In a tree structure, the instance John's-sailboat inherits all slots and facets from Sailboat-frame, Boat-frame, and Vehicle-frame. In a lattice, each frame inherits from multiple ancestral lines.

ure 4). Gymnosperm-frame contains slots that describe coniferous trees in general. Tree-in-my-backyard contains slots that describe a particular tree.

Frames represent knowledge about an object, concept, event, or problem. They can be more powerful than conventional record structures when they are arranged in trees or lattices because they can take advantage of the hierarchy to inherit slots and facets from other frames.

A frame at the root of a tree represents a concept or object at the highest level of abstraction. Frames at the leaf nodes of the tree represent actual instances. The higher the level at which a frame is located in the tree, the less specific its knowledge. Frames inherit the slots of their ancestors. In this way, a leaf frame can inherit the slots of every frame on the unique path between it and the root frame.

A partially completed frame tree containing knowledge about vehicles is shown in figure 5. The root frame is Vehicle-frame, and its children are Airplane-frame, Boat-frame, and Car-frame. Each child contains all its parent's slots. Sailboat-frame and Motor-boat-frame each contain the slots of Boat-frame in addition to their own unique slots. An instance of Sailboat-frame called John's-sailboat would contain not only the slots rig, steering, sail-area, but also length, medium, and every other slot in its ancestor frames.

Inheritance is a powerful knowledge representation method. It eliminates the need to store information that can be seen in context. It allows inferences to be made about the structure of problems and concepts, and the relationship between their parts. Facts (assertions) that can be inferred from the frame tree about a sailboat are:

- A sailboat is a type of machine.
- A sailboat is a type of vehicle.
- It moves through the water.
- It has a maximum speed.
- Its displacement is related to its construction material.
- It has one or more masts.

Frames arranged in a graph or lattice can inherit slots from multiple frames elsewhere in the lattice, not just from one ancestral line as in trees. For this reason, lattices are suited to complex inheritance.

INFERENCING WITH FRAMES

Frames can be involved in inferencing in several ways. One method is called expectation-driven inferencing, which is illustrated at a high level by the recursive algorithm Satisfy (see figure 6). Satisfy attaches values to each slot in a frame by executing procedures, testing assertions, firing rules, or satisfying child frames. If all slots can be filled, the part of the problem represented by the frame is solved. If a slot requires satisfaction of a child frame, then Satisfy is called recursively.

This algorithm can be demonstrated using the frame tree in figure 5. To confirm an object is a vehicle and to determine what type, a new frame called Mystery-object is instantiated. Slots and values are added by examining generic frames. Beginning with Vehicle-frame, inference proceeds in the context of a vehicle. If the object can be shown to meet all expectations for a vehicle, it is confirmed. If not, the object must be examined as potentially belonging to another context (for example, toy). If no further contexts are available in the knowledge base, the object cannot be defined without adding information to the base.

To satisfy Vehicle-frame in figure 7, the medium slot must be satisfied first. Because its value is really an assertion, it can be satisfied by the function test-assertion, which checks the If-needed facet and finds that the user must be queried for the information. Developers can include a query string such as "What is the medium in which the object is used?" or "Is the object used in water?" to be presented by the user interface. If the user gives a satisfactory response, such as water, the process continues with the next slot. Type is the goal slot, which contains the names of frames. Because an actual frame name is needed for the recursive call to Satisfy, the If-needed facet is triggered and provides the value Boat-frame. Satisfy is then called recursively to attempt to satisfy Boat-frame.

When a frame is satisfied, in this case Boat-frame, the value *true* is returned; the Vehicle-frame is also satisfied and the algorithm can terminate. If Boat-frame cannot be satisfied, an alternative to Satisfy can be used for inferencing. For example, the inference engine could assume that another type of water vehicle exists. The engine might then attempt to analyze the differences between the unidentified object and a sailboat and might ultimately create a new frame.

The engine might conclude that the user is incorrect in stating that the vehicle operates in water because it is inconsistent with the user's other observations and the system's knowledge of boats and vehicles. The system could also backtrack, assuming the object is not a vehicle, and explore the problem from a different context. When all else

fails, the inference engine might conclude there is inadequate knowledge to identify the object.

For cases in which the identity of an object is known and details about it are requested, yet another inferencing method can be used. If object *X* is a sailboat, an instance of Sailboat-frame called Sailboat-*X* can be created to hold knowledge of the object. The inference engine can ask questions about its displacement, length, speed, type of keel, number of masts, and so on. It will not need to ask about the medium and method of propulsion because this can be inferred from the knowledge base.

When an object is known to have masts and a wheel, an alternative inference method can be used. A simple search of the knowledge base could find all frames containing slots that are consistent with known assertions. Then Satisfy could be used to identify one or more feasible contexts, such as Sailboat-frame, in which to examine the object. The inference engine could try to fill all necessary slots by asking questions appropriate to a sailboat. No standard inference method yet exists for frames, in part because they are often combined with other knowledge-representation methods.

Frame-based knowledge representation has several advantages. New concepts and objects can be recognized faster and classified in terms of known concepts and objects as in human reasoning. New items inherit slots from analogous situations so that different slots show contrast between old and new concepts, and shared slots represent similarities.

Expert systems using frames cope with missing information in a problem description better than rules do. Another strength is that a single unit of knowledge represented by a frame can be interpreted in different contexts so that new frames rarely affect existing frames in knowledge bases.

Frames are only a knowledge organization method and their power depends on the specific inferencing method. They are often embedded in, and supplement, other knowledge representations. For example, frames partition a rule base by attaching rules to slots along with other information contained in the hierarchical frame structure or by storing knowledge in nodes of a semantic network.

Frame-based representation is appropriate for problem domains in which knowledge is hierarchical or contextual. Such domains include plant taxonomy, language interpretation, political analysis, and power-plant simulation. It is also appropriate for domains with missing or conflicting knowledge.

FIGURE 6: ***Satisfy***

```
Function SATISFY (Goal : Slot-type) : Boolean;
Begin
SATISFY := true;
  Case (Type(Goal)) of
  Frame:
    For a-slot := first-slot-in-frame TO
                  last-slot-in-frame
      IF NOT(SATISFY(a-slot))
      THEN   SATISFY := false;
  Procedure: SATISFY := can-do-procedure(Goal);
  Rule:      SATISFY := test-rule-conditions
                        (Goal);
  Assertion: SATISFY := test-assertion(Goal);
  .
  .
  .
  end
end; /* SATISFY */
```

Satify, an example of expectation-driven inferencing, attempts to attach values to each frame slot by executing procedures, firing rules, testing assertions, or satisfying child frames.

Frames can be used to solve problems requiring knowledge of chronologically distinct events, which is extremely difficult to do with rules. Generic frames, called *scripts*, can specify sequences of events expected in specific contexts. Just as an inference engine can verify existence of expected assertions, it can also confirm that events (represented by an instance frame) have taken place.

CONTROL STRATEGIES

Whereas expert systems are best categorized by knowledge representation methods, inference engines are best defined by control strategies, which determine the order of inferences. Two strategies in most expert system shells are forward chaining (data-driven) and backward chaining (goal-driven).

Forward chaining starts with assertions about a problem, makes all inferences possible, and draws conclusions. No prior knowledge is available to predict what inferences can be made using the knowledge, or whether a particular conclusion can be inferred from available information. It is most appropriate when all knowledge needed to make a decision is available before inferencing begins and when fewer final conclusions exist than initial assertions.

Figure 8 shows a simple algorithm for forward chaining. Every rule in the rule base is checked sequentially to see if it can fire. After each rule has been checked, the process begins again. It ends only when every rule has been checked in sequence, no more rules fire, and either a particular goal is reached or no further changes occur to the knowledge base. The following list summarizes the inferencing that occurs when forward chaining is used on the personnel rule base shown in figure 2:

RULE	CONCLUSION
1	Tom is late for work
0	Tom's car broke down
2	Tom has good excuse for being late
7	Tom sold at least 1 million units
6	Tom is a good employee
3	Tom will not be dismissed

Backward chaining, on the other hand, starts with an answer and works back to the problem description. Rule selection is guided by conclusions rather than conditions. An answer is proven correct if the system can infer initial assertions that coincide with the initial problem situation. Although the concept of backward chaining is relatively simple, the Pascal-like code needed for it is complex (see figure 9). If the system finds an assertion that matches the goal, then the goal is proven. If no match is found, the system looks for a rule whose conclusion contains the goal. If every condition in the rule is proven true, then the goal is proven. Otherwise, the process continues recursively on other rules until a match is found.

Inferences made by backward chaining can be traced using the personnel rule base in figure 2. The goal is to show that "Tom will not be dismissed." To do this, the system must establish that "Tom is a good employee" and "Tom has good excuse for being late." The first assertion can be concluded only if one of the following can be proven: "Tom is the boss's son-in-law" (Rule 5) or "Tom sold at least 1 million units last year" (Rule 6).

Because the assertion base has no knowledge that "Tom is the boss's son-in-law," the system cannot make this inference. If the system can show that "Tom vacationed in St. Croix last year," then it can prove that "Tom sold at least 1 million units last year." It can then infer that "Tom is a good employee." The system still must prove that "Tom has good excuse for being late." It does this by using the assertion base to match Rule 0, then Rule 2.

Backward chaining can be more efficient than forward chaining; if the starting goal is correct, less time is spent making inferences that lead down blind alleys. Backward chaining is appropriate when the user makes a good guess about a possible solution

FIGURE 7: *Detailed Frames in the Vehicle Hierarchy*

```
NAME:              VEHICLE-FRAME
SPECIALIZATION-OF: MACHINE-FRAME
PURPOSE:     "Conveys people from one place to another through a medium"
  MEDIUM:
    RANGE:     (AIR, WATER, LAND)
    IF-NEEDED: Ask user
    DEFAULT:   LAND
  VEHICLE-TYPE:
    RANGE:     (AIRPLANE-FRAME, CAR-FRAME, BOAT-FRAME)
    IF-NEEDED: IF MEDIUM = WATER THEN BOAT-FRAME
               IF MEDIUM = AIR THEN AIRPLANE-FRAME
               OTHERWISE CAR-FRAME
  MAXIMUM-NUMBER-OF-PASSENGERS:
    RANGE:     an integer
    IF-NEEDED: Ask user
--------------------------------------------------------------
NAME:              BOAT-FRAME
SPECIALIZATION-OF: VEHICLE-FRAME
  LENGTH:      Real number
    IF-ADDED:  Recalculate-maximum-speed(LENGTH,LOW)
  BEAM:
    RANGE:     Real number
  DISPLACEMENT:
    RANGE:     Real number
    IF-NEEDED: Ask user
  MATERIAL:
    RANGE:     (Fiberglass, wood, concrete, metal)
    IF-ADDED:  Recalculate-displacement(LENGTH, BEAM)
  TYPE:
    RANGE:     (SAILBOAT-FRAME, MOTORBOAT-FRAME)
  PROPULSION:
  DRAFT:
    RANGE:     A real number
    IF-NEEDED: Calculate-draft(LENGTH, DISPLACEMENT, TYPE)
```

```
--------------------------------------------------------------
NAME:              MOTORBOAT-FRAME
SPECIALIZATION-OF: BOAT-FRAME
  PROPULSION:
    RANGE:     (Diesel, gasoline, methane, ammonia, steam, electric)
    IF-NEEDED: Ask user
  MODEL:
    IF-NEEDED: Ask user
  HULL:
    RANGE:     (Skiff, Pram, Utility, Runabout, Cruiser, Other)
    IF-NEEDED: Ask user
--------------------------------------------------------------
NAME:              SAILBOAT-FRAME
SPECIALIZATION-OF: BOAT-FRAME
  RIG:
   RANGE:      (Schooner, Sloop, Lanteen, Yawl, Ketch)
   IF-NEEDED:  find-type-of-rig(NUMBER-OF-MASTS, LENGTH)
               Ask user
  NUMBER-OF-MASTS:
   RANGE:      (1..6)
   IF-NEEDED:  find-number-of-masts(RIG, LENGTH)
               Ask user
  MODEL:
   IF-NEEDED:  Ask User
  STEERING:
   RANGE:      (Wheel, Tiller)
   IF-NEEDED:  Ask User
   DEFAULT:    If LENGTH > 28 Then Wheel, ELSE Tiller
  KEEL:
   RANGE:      (Centerboard, Dagger, Fin)
   IF-NEEDED:  Ask User
  SAIL-AREA:
   RANGE:      A real number
   IF-NEEDED:  calculate-sail-area(DISPLACEMENT, LENGTH, RIG, NUMBER-OF-MASTS)
```

The root (Vehicle-frame) has the most general characteristics. Frames further down inherit all characteristics from frames in their ancestral lines. One method of inferencing with frames involves trying to fill slots to identify or characterize an object.

and when more goals exist than combinations of initial assertions. Generating natural language text often is easiest with backward chaining.

Medical diagnosis is an appropriate backward chaining situation. Physicians often make good guesses about the cause of a disease, allowing an expert system to predict observations. Backward chaining is ideal for robot planning. With a final configuration of objects as the goal, an inference engine can backward chain to determine the movements needed to achieve it.

Forward and backward chaining combined in an expert system shell provides more flexibility to users. Forward chaining determines what can be inferred, while backward chaining determines what needs to be inferred.

EXTENDING THE OPTIONS

Besides the inference mechanism itself, the inference engine can contain optional components such as explanation, knowledge acquisition, and knowledge-base maintenance facilities.

An explanation component gives an expert system the ability to show how it reaches a conclusion. Explanations can be as simple as displaying rule-fire sequence or as sophisticated as maintaining a separate database of knowledge to reason about why certain inferences are made. Explanation output can be filtered through the natural language interface.

Explanation systems are critical because few experts will take the advice of a system that cannot justify its conclusions, especially in fields such as medical diagnosis and treatment. A good explanation system turns an expert system into a refined teaching tool that provides nonexperts with detailed explanations of its reasoning in hypothetical problem-solving situations.

Domain experts can use knowledge acquisition to incorporate facts into the knowledge base. Simple shells provide only prompts for rule components, while sophisticated ones have graphical knowledge-base displays and can reason about information needed in the knowledge base.

Knowledge-base maintenance helps build and maintain large knowledge bases, often with both automatic and interactive functions. Automatic functions check and flag inconsistent and redundant rules and inconsistencies between knowledge-base components; interactive functions provide a graphical knowledge-base editor to remove and modify knowledge, and facilities for testing parts of the knowledge base with the inference engine. Overlap often exists between acquisition and maintenance tools.

RESEARCH PATHS

Expert system research is geared toward achieving greatest productivity with existing technology and developing new technology. Better tools are the way to decrease development time, increase knowledge-base accuracy, and simplify user interfaces. The most successful tools allow creation of hybrid structures; some are even custom designed. The next generation of shells might hide implementation details instead of requiring developers to choose between knowledge representation features.

Integrating expert systems with a wider variety of databases and programs is another goal. This includes generating all or part of a knowledge base from existing conventional databases because they are usually broad in scope, centralized, timely, and easily accessible throughout an organization.

FIGURE 8: *Forward-chaining Algorithm*

```
Change := false
REPEAT
  FOR a-rule := first-rule TO last-rule DO
    BEGIN
      IF    Test-conditions(a-rule, Knowledge-Base)
      THEN  BEGIN
        Do-actions(a-rule, Knowledge-Base);
        Change := true
        END
    END
UNTIL not(Change) OR goal-is-found(Knowledge-Base);
```

In this algorithm, if the antecedent is satisfied, the conclusion is placed in the assertion base. This continues until no new assertion can be made or the problem is solved.

FIGURE 9: *Backward-chaining Algorithm*

```
Identify the goal (or subgoal)
IF the goal is a known assertion,
   THEN it can be inferred.
   ELSE Look for a rule whose conclusion contains the goal.
      IF one does not exist,
      THEN the goal cannot be inferred.
         ELSE (one does exist)
         IF every condition in the antecedent (subgoals)
            can be inferred,
         THEN the goal can be inferred
         ELSE look for another rule.
```

Backward chaining is initiated when a goal is specified. The system works back to the problem description by matching the goal and any subgoals generated to rule consequents.

Spreadsheets, a primary source of raw data on most PCs, already are accessible to some shells. Natural language interfaces will continue to improve so that novice users can communicate more freely with the completed system.

Shells and delivery systems already are migrating from specialized hardware and large mainframes to PC-based environments. Shell vendors will continue this trend because it opens the door to a larger user base. It should also mean they will develop advanced shell and mainframe communications in the PC environment.

Design and construction are other arenas for technological progress. These include recent distributed models of knowledge representation and inference control that, inspired from the architecture of the human brain, distribute knowledge throughout a network of simple structures. This kind of distribution reduces search time for specific knowledge and decreases potential damage from incorrect or missing knowledge.

Efforts have been under way for many years to give systems the ability to learn by discovery or by studying analogies or examples. Experimental frame-based systems can understand natural language stories and draw conclusions about actions similar to those in the story. Frame-based systems have also been used for studies of learning through analogy.

ETHICAL ISSUES

The increasing use of expert systems in commercial and social applications brings with it a number of ethical questions. These are similar to the ones asked 40 years ago when some people feared that computers would dehumanize society.

Many people now fear that widespread introduction of expert systems in the workplace will make experienced (and higher-paid) personnel more expendable. They might also lessen incentive for younger workers to develop experience and knowledge about their jobs.

As more people contribute to the complexity of expert systems and rely on their decisions, responsibility for decisions will become more diffused. Who will be morally responsible for death resulting from an incorrect medical diagnosis by an expert system: the attending physician who took its advice, the original domain expert, the knowledge engineer, the programmer, or the tool designer? Of seemingly greater practical importance in our society is the question, Who is legally responsible? No simple answers exist.

Assuming that future expert systems technology will permit the embodiment of almost any expert human skill in software, where will the line be drawn? Should our society create a place for expert systems as priests, generals, doctors, judges, social workers, teachers, news writers, and politicians, to name a few?

Despite ethical concerns and technological growth, current expert systems are the best bet that many businesses and organizations have for solving difficult problems—especially those requiring heuristic knowledge, those with conflicting or uncertain data, and those with large amounts of input.

Shells simplify and improve transfer of knowledge from a human expert to a ready-made knowledge structure; the most sophisticated ones combine rules, frames, and procedures. A shell also should have access to a standard artificial intelligence programming language (such as LISP or Prolog) so that knowledge can be represented using procedures. Shells allow incremental development of large systems, rapid prototyping, and delivery on systems other than the development system. Good shells are critical to efficient development of real-world expert systems to solve real-world problems. They can make the difference between success or failure.

SUGGESTED READING

Barr, A. and E. Feigenbaum, *The Handbook of Artificial Intelligence*, Vol. I, William Kaufman Inc., 1981.

Charniak, E., and D. McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley Publishing Company, 1985.

Citrenbaum, R., J. Geissman, and R. Schultz, *Desirable Characteristics for Expert System Shells*, Proceedings of the Third Annual Artificial Intelligence and Advanced Computer Technology Conference, Abacus Programming Corporation, April 1987.

Davis, R., *Knowledge-based Expert Systems: Planning and Implementation*, Addison-Wesley Training Systems, 1987.

Davis, R. and D. B. Lenat, *Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill Inc., 1982.

Harmon, P., and King, D., *Artificial Intelligence in Business*, John Wiley and Sons Inc., 1985.

Leininger, D., *Expert System Environment for Application Development: Criteria for Selecting a Tool*, Proceedings of the Third Annual Artificial Intelligence and Advanced Computer Technology Conference, MDBS Inc., April 1987.

McDermott, J., "R1: A Rule-Based Configurer of Computer Systems," *Artificial Intelligence*, Vol. 19, No. 1, 1982.

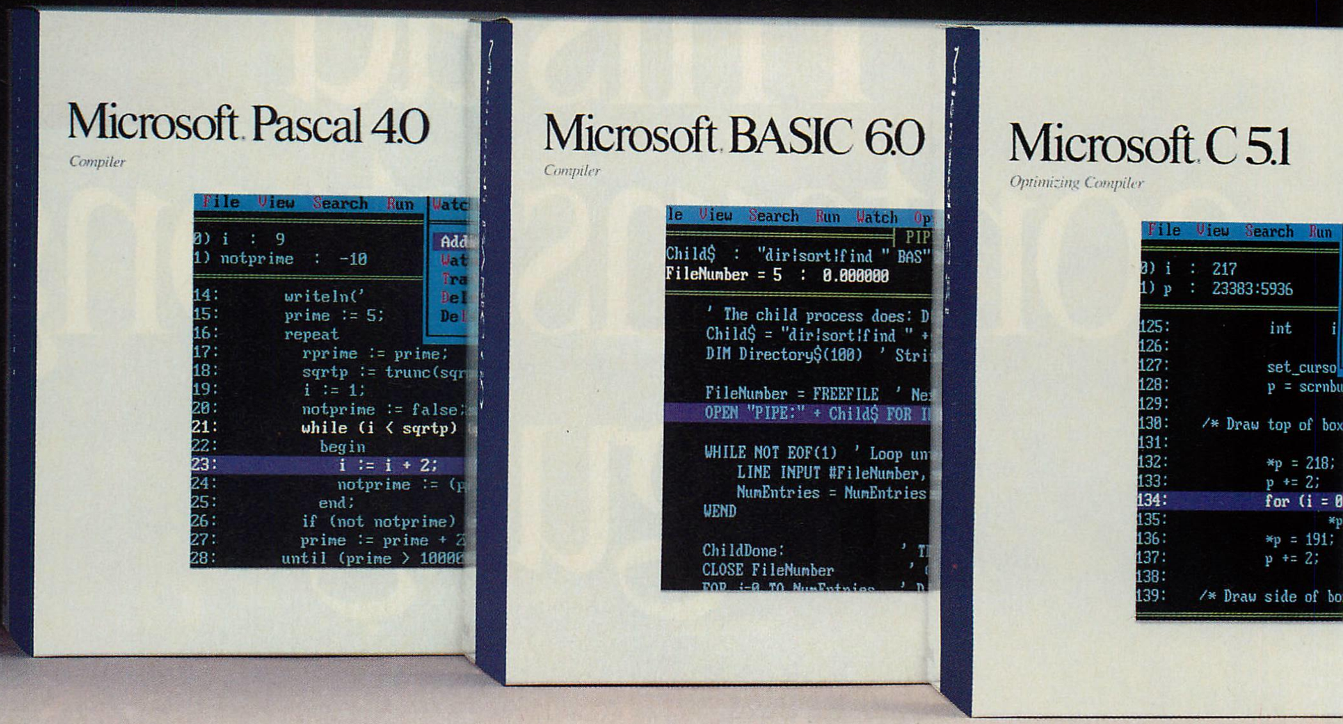
Winston, P. H., *Artificial Intelligence*, second edition, Addison-Wesley Publishing Company, 1984.

Tom Arcidiacono is an instructor of computer science at the New York Institute of Technology and a consultant in knowledge-based systems development.

WARNING:

This ad contains strong language.

Introducing Mic
for OS/2



The people who co-developed the industry's most powerful personal computer operating system are now proud to announce programming languages to match.

Introducing Microsoft® Macro Assembler 5.1, C 5.1, Pascal 4.0, FORTRAN 4.1 and BASIC Compiler 6.0.

Five industrial-strength, stand-alone languages that combine the implementation flexibility you've enjoyed under MS-DOS® (which, of course, they still support) with the advanced capabilities you've anticipated from OS/2.

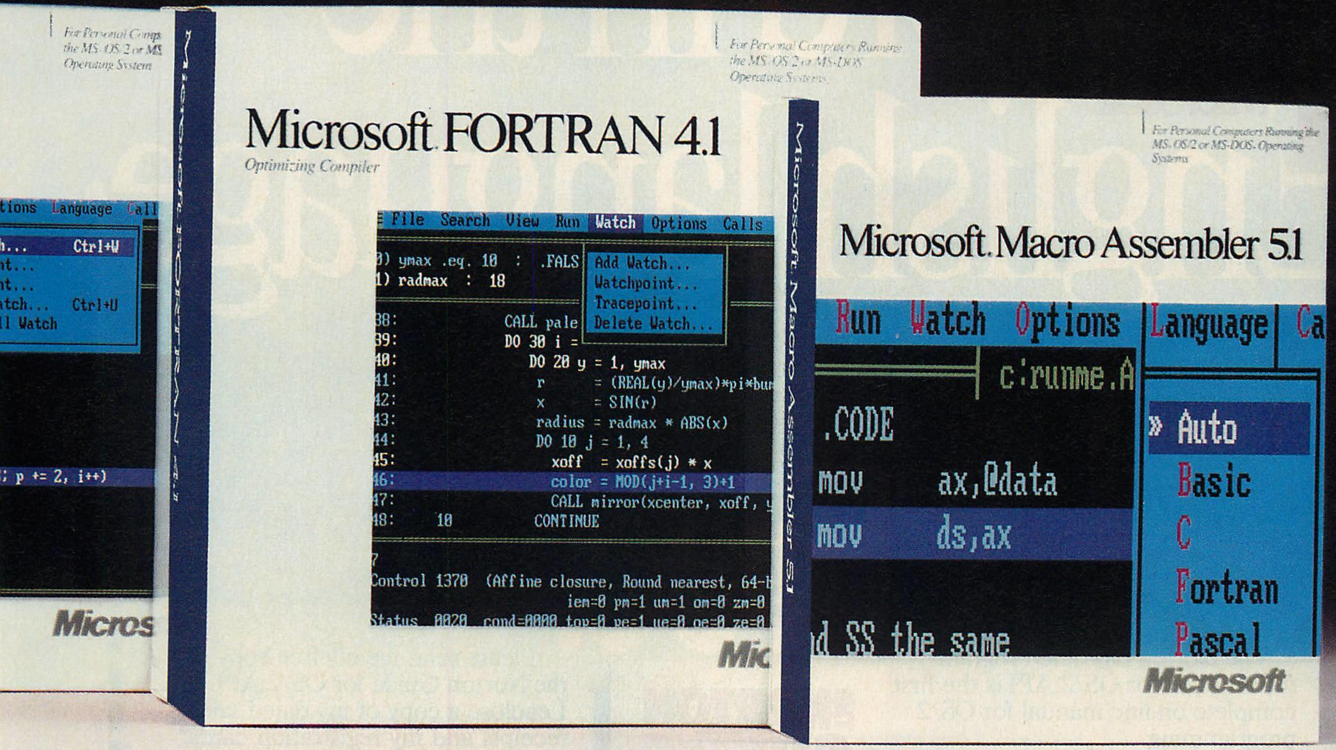
Capabilities such as the ability to develop large, sophisticated applications which go beyond the 640K barrier, taking advantage of up to 16MB of RAM, and utilizing the potential of today's microprocessors.

Just like their MS-DOS predecessors, these five new languages are equipped with powerful, professional features you work with, not around:

Support of direct calls to the operating system, and inter-language calling for mixing multiple languages on the same project.

Access to OS/2 system calls and a full complement of utilities, including an incredibly fast incremental linker and the

rosoft Languages
systems.



first protected mode programmer's editor that works equally well in real mode.

Microsoft CodeView,® our popular, advanced debugger that lets you untangle program logic at the source code level, no matter what code you're using.

(It even lets you debug protected mode programs up to 128MB of virtual memory, and larger programs than ever before in real mode.)

As the perfect complement to our new languages, we're also offering the Microsoft OS/2 Programmer's Toolkit.

It contains a parameter-by-parameter breakdown of all OS/2 system calls and samples to get you started.

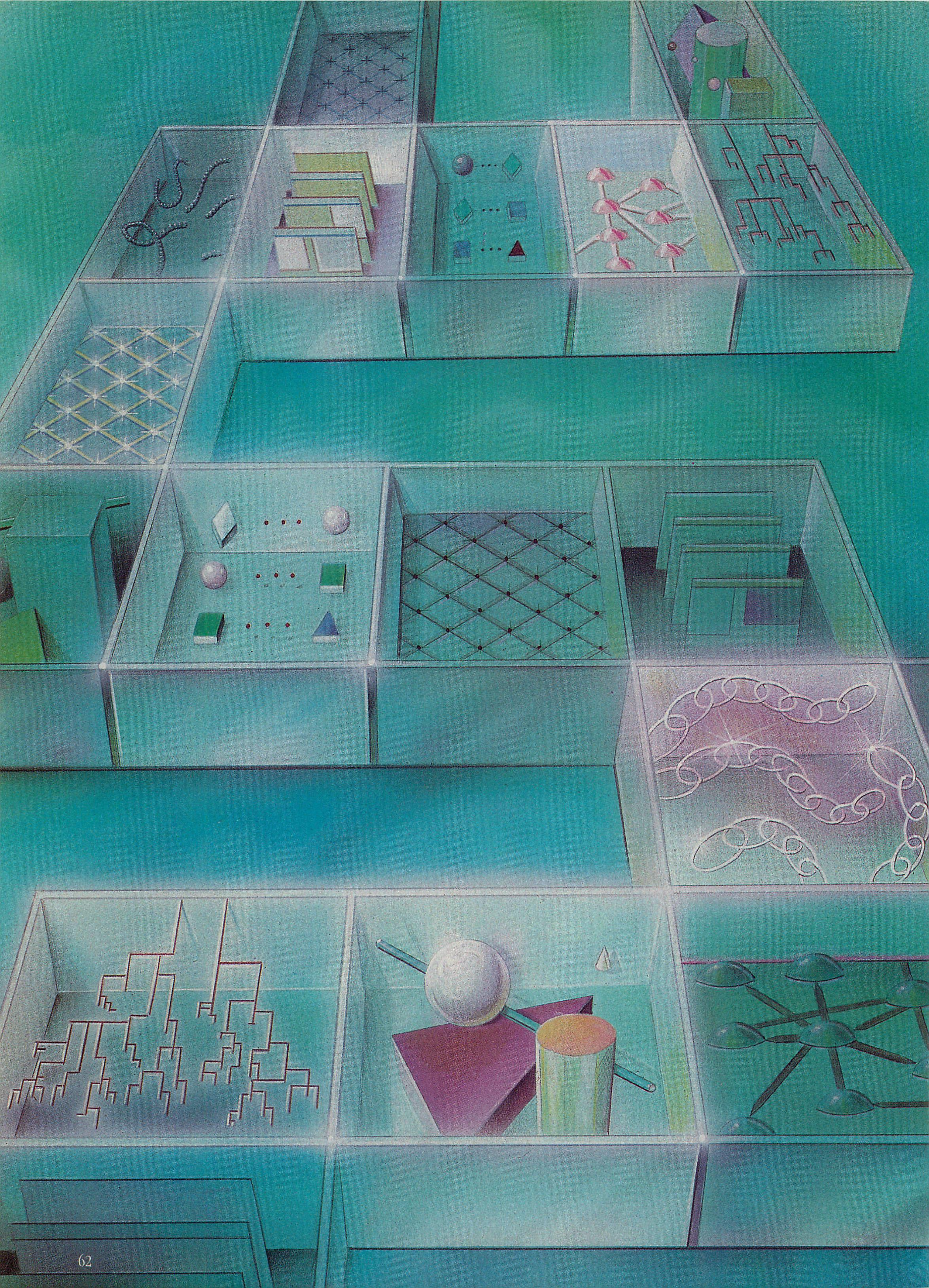
All the tools you need for turning out larger, more powerful, more complex OS/2 applications.

(And, incidentally, all the tools we rely on for creating our own commercial software.)

For the name of your nearest Microsoft professional languages dealer, simply call 800-541-1261, Dept. C15.

Ask him for some more information on our OS/2 family.

He'll show you some languages you can really swear by.



Elements of Expert System Shells

Expert system shells offer a variety of approaches to solving a problem. Choosing the best course requires careful evaluation of all parts of the shell.

MAXINE FONTANA and JORDENE ZEIMETZ

Once the realm of researchers and academicians, expert systems are advancing into the business sector to solve real-world problems. Yet, few managers and software developers know exactly what expert systems are or how they work.

To familiarize developers with these systems, *PC Tech Journal* begins in this issue a series of reviews of expert system shells, the tools used in building expert systems. The first review, an evaluation of Gold Hill's GoldWorks, begins on page 68.

The editors have established criteria for evaluating these shells. The criteria will cover shell, based on rules and frames, currently the most common structures used by expert systems are based (see "Computerized Reasoning," Tom Arcidiacono, this issue, p. 44). Many shells, including GoldWorks, combine rules and frames.

Shells for the PC, which range in price from $99 to $7,500, are made up of many generic features that developers can exploit to create expert systems for delivery to business, medical institutions, industry, and so on. These features include a knowledge representation method (such as frames or rules); an inference engine and control strategies (including backward and forward chaining) that guide the system's inferencing processes; a user interface; explanation and maintenance facilities; and interfaces to other programs such as graphics and data managers.

Unlike completed expert systems, shells are not domain-specific—that is, they can be used to solve problems in a variety of different disciplines. Just as varied problems require different reasoning processes by human experts, they also require different constructs of tools. One reason that shells are so diverse is that the representation models and inference engines at their heart are so varied in complexity and capacity.

Setting criteria for evaluating expert system shells is complicated by this variety of shell design. Highlighting the individual parts of shells is the best way to differentiate among them.

Applying benchmarks, which measure the time required to perform a specific task, will not be done. For expert systems, the soundness of a solution ultimately is more essential than how fast it runs the race.

Instead, each reviewed shell will run a sample application tailored to that specific shell. While all expert system applications are aimed at solving a problem, the amount of knowledge, solution desired, and complexity vary.

Selecting the appropriate shell to tackle a specific problem means understanding the theory and technology

ILLUSTRATION • ANDY LEVINE

behind the shell. Our evaluation criteria are designed to ask the right questions about each shell. What follows is a sampling of questions we will answer in reviewing expert system shells.

The user interface is one of the most noticeable features of any software package. For a menu-driven interface, are the main menu and submenus easily navigable? For a command-driven interface, is the command syntax easy to remember, tolerant of errors, quick and convenient to use? Is there a programmable interface for customization?

Does the shell provide developers with tools to paint screens and menus? Does it offer a full-screen editor appropriate to the shell's language, or does it allow the user to select an editor?

A large-scale project might require data to be incorporated from databases, spreadsheets, and other files into the shell. Therefore, can the shell access DOS and system functions, data from other sources, or other programs such as data managers or spreadsheets? Can it call programs written in other languages (such as LISP or C) to write custom interfaces?

The shell must accommodate the fact that this technology is still new and unknown to most developers. In that vein, does the shell offer a complete tutorial? Does it assume prior knowledge of expert systems or does it start from scratch? Is the sample problem trivial or is it representative of a typical user problem? Does it reveal intricacies of the shell?

Does the documentation provide a wide range of examples? Does it detail the theoretical basis of the product? Does it describe the types of problems best solved by the product?

KNOWLEDGE REPRESENTATION

To be able to manipulate shells to their advantage, developers must understand the significance of the knowledge representation methods used: *assertions* (facts), *rules* (if . . . then statements), and *frames* (structures having slots to hold data and procedures, and forming a tree or lattice to allow inheritance between data). The sample application will illustrate the use of assertions, rules, and frames.

Shells differ in the utilities they provide to maintain knowledge. As the knowledge base grows, can the user update it without the developer's help?

Can rules and frames be commented? Can the shell deliver rules and frames as output showing a tree or hierarchy indicating the line of reasoning? Does the shell update related knowledge as new facts are entered? Can the knowledge base be inspected in detail at any time?

Assertions. Some shells allow developers and users to enter assertions directly from the keyboard, some only generate assertions automatically when rule conditions are met or slots in frames are filled, and others support all methods. Do assertions have to be entered in a special format? If so, is a template provided? What types of assertions are supported? Is there a method to control the contents of the assertion base so that retraction (deletion) of an assertion causes retraction of other assertions that depend on it?

Rules. A shell's ability to structure rules can be judged by asking how rules are entered—using a template, full-screen editor, or menu selection. What is its syntax? Are meta-rules (rules about rules) supported?

Some enhanced features of shells include optimizing the sequence of rules entered for future search efficiency and validating data on entry. What steps are taken to provide consistency and completeness in the knowledge base?

Frames. A frame-based shell should provide developers with an easy-to-use environment. Are frames entered via a menu, editor, or commands? How many slots can a frame have? What type of information can the slot hold: numerical values, character strings, procedures? Can procedures be attached?

Many slots in frames contain facets, which store knowledge about the information in slots. The knowledge might be related to documentation and explanations, restrictions on slot values, defaults, confidence factors, and procedures to be executed when the slot is modified. Which facets are supported?

Inheritance is important for solving complex problems. How is inheritance set up? Is multiple inheritance supported? How are slots and facets inherited (tree, lattice, or other)?

INFERENCE ENGINE

The degree of versatility and flexibility that inference engines offer developers and users is a consideration. Are forward chaining, backward chaining, and an integrated combination supported? How does each work? What types of problems does it solve? If multiple strategies are supported, why is a combination important?

Are rules searched by order of entry, priority, or certainty? When multiple matches are found, how does the shell choose the one to be completed first? Some shells choose the first match, while others offer more flexibility, such as prioritizing.

How does the engine search through the knowledge base? Can subsets of rules be searched? Can the user modify or fine-tune search strategies? Does the shell interface directly with a database, spreadsheet, or user to retrieve data values? How does the shell work around missing or incomplete data? Can it answer what-if questions?

Because many expert systems can produce more than one reasonable answer, the selection process is important. Some shell mechanisms select the best out of a series of reasonable answers or ask the user for input to help the system select the best answer. Others present one answer, and still others offer many solutions, allowing the user to choose one. In a nuclear power plant, decisiveness and speed might be the most important factors, while a doctor might prefer to choose among alternatives for a medical diagnosis.

TRACING AND OTHER TREATS

Some shells provide audit trails so the user can check every step the system takes in reaching a solution. Is the end user provided with a trace of every action of the inference engine: the current goal, the rules being searched, the rules that fired and why? Can the trace be listed or observed on the screen?

Many shells have the ability to explain reasoning. Does the developer have to enter English translations for each rule in the explanation facility or does the shell generate them? Are the explanations updated automatically when rules are changed or assertions are retracted?

In terms of performance, does the shell solve the given problem in a reasonable amount of time? Is modular design supported? Does the shell have the ability to learn?

The expert system developer must compare many competing shells to pick the one best suited to a problem. The product's knowledge representation method, inference engine, user interface, and special features should be examined early and compared with other shells. This means being well equipped to get the most out of a shell and knowing its strengths and limitations. If the answer pursued is "The best solution," then the question asked should be "Which shell?"

Maxine Fontana is a technical editor and Jordene Zeimetz is an associate editor for PC Tech Journal.

For 500 of Us, They Will Come on June 13-15. SYSTEMS FORUM 88.

Confusion. Conflict. Chaos. These are the spirits that create the questions of today's systems issues, and that make it tougher for you to predict what's going to happen tomorrow.

But in June, some of us will know more than the rest of us about what is happening now and what is going to happen.

June 13 - 15 are the dates for the second annual PC Tech Journal Systems Forum in San Francisco. This is where industry leaders -- both vendors and users -- will gather to handle the industry's hottest topics, and to take some of the confusion, conflict and chaos out of all that is happening to PC systems today.

Last year, the first PC Tech Journal Systems Forum illuminated and impressed. The systems professionals who attended came away with the most expert and current information available. Some comments ... "One of the most relevant forums/conferences I've been to," and "PC Tech Journal did a great job. I especially like Will Fastie's Donahue approach as moderator."

And for 500 systems professionals, the PC Tech Journal Systems Forum 88 will be even more illuminating than in 1987. This isn't a lecture series; it's expert panelists and a professional audience, posing questions and seeking answers for business. You'll have your chance to explore the issues and problems that concern you most. Here's a sample of what you'll encounter at PC Tech Journal's Systems Forum 88.

1. OS/2: Out of the Starting Gate. Now that it's out, where is it going? What is the state of software for OS/2? Do we have the right tools -- and enough of them?

2. Toward the 386 Platform. Is the 386 just a faster 286? Or is it opening up new desktop applications? What will falling 386 prices do to the sale of 286 and 8088/86 machines? Will a 386 add-in board give you a full-fledged 386 machine?

3. Bus Wars. Who will win the war for bus supremacy -- IBM's Micro Channel, the "classic" PC bus, or AST's new SmartBus? Will one emerge as the standard? Which one? What does that mean to you?

4. Meeting the Needs of the End User. Today, with centralized control taking the "personal" out of "personal computer," what do end users need? How do they get it? How are other important issues, such as training or software updates, handled?

5. SQL: A Midterm Report. Now that it appears to be the standard, what must systems managers know about SQL to deploy distributed systems? Is SQL the only answer?

TECH PC JOURNAL
Systems Forum '88
SAN FRANCISCO

6. Developing Applications for the Graphical User Environment. The Macintosh is here and the Presentation Manager is on the way. Are these new environments the basis for new solutions or more problems? Do the environments provide enough function for the typical developer?

7. At the Desktop: A Hardware Melting Pot. What software and systems issues are created by the typical melange of IBM compatibles, workstations and Macintoshes found in the workplace?

8. Managing Your Network. In the growing world of LANs, how does the manager plan for administering them now and project resources necessary to manage them over time?

9. The Network API Supermarket. With all of the application programming interfaces (APIs) available today, which one should the developer choose? How much is the choice affected by the network software? Is there a standard? And if not, do we need one?

10. Networking Networks. If managing one network is a challenge, how do you manage a network of networks? How are they connected? How are the hardware and software in each LAN integrated with the others?

The PC Tech Journal Systems Forum is a working meeting -- a place where professionals learn from professionals. Because of that, registration is limited to 500.

The cost for the three-day forum is $ 795 ($ 695 if you register before April 15). Ask for special PC Tech Journal rates at the Westin St. Francis, the official forum hotel, by calling 415-397-7000 or Westin national reservations at 800-228-3000.

Register now, and get ready for the most enlightening and most important PC meeting of the year -- beginning June 13.

To register or for further information call

800-544-PCTJ.

The Age of GoldWorks

As expert systems mature into viable applications for the PC, GoldWorks will be at the forefront. This expert system shell will make its name developing upscale mainframe-type applications.

KEN LEVINE

Some products are as good as gold, while others just pretend to be. GoldWorks 1.1, from Gold Hill Computers Inc., does not have to pretend—it comes very close. In both price ($7,500) and capabilities, GoldWorks is a high-end expert system shell for the PC. GoldWorks provides sophisticated, flexible features that achieve near-mainframe quality, but it is surprisingly easy to use.

GoldWorks combines rule- and frame-based knowledge representation methods with integrated forward and backward chaining, and it includes object programming, which allows the developer to attach LISP functions to frames. Unique architectural features include object programming, sponsors and agenda-based processing, and user-defined certainty calculus.

GoldWorks' layered design (see figure 1) makes it appropriate for both inexperienced and veteran developers. The top layer consists of a menu-driven user interface, a developer's interface, and interfaces to Lotus 1-2-3, dBASE III PLUS files, and user programs written in Microsoft C or Lattice C. Less-experienced developers can work directly with this top layer, which requires no LISP programming experience. For beginners or developers not familiar with a LISP-based environment, GoldWorks provides an excellent on-line tutorial and detailed, well-written documentation so they can quickly become expert at using this complex package.

The middle layer consists of the knowledge base of GoldWorks objects (such as frames, rules, assertions), and the inference engine, which supports forward chaining, backward chaining, and goal-directed forward chaining (integrated forward and backward chaining). The bottom layer consists of Golden Common LISP (a large subset of Common LISP) to write LISP routines. It includes GMACS, a powerful full-screen editor that has most or all features expected in an EMACS-style full-screen editor (such as multiple windows and buffers, kill ring, tag tables, definition formatting, incremental interpretation, and compilation within the editor itself).

The experienced developer can use the many features that are available at these lower layers to build systems with more sophisticated features, such as user-defined certainty reasoning and object programming. The developer's interface provides advanced users with a frame-based screen toolkit with which they can create custom menus, windows, and screens.

GoldWorks runs on the IBM PC/AT and Compaq Deskpro 386 (and 100-percent compatible systems) as well as the Hummingboard (an add-on board for a PC/XT or PC/AT designed by A.I. Architects and distributed by Gold Hill Computers). The Hummingboard is a 16- or 20-MHz 80386 with 6MB to 24MB of memory that is designed to run LISP and GoldWorks in lieu of the XT or AT processor.

The product runs in extended memory, except for a small kernel that runs in base memory. Memory requirements are 512KB of base memory (leaving the rest for terminate-and-stay resident, or TSR, programs and user-written code). Non-Hummingboard sys-

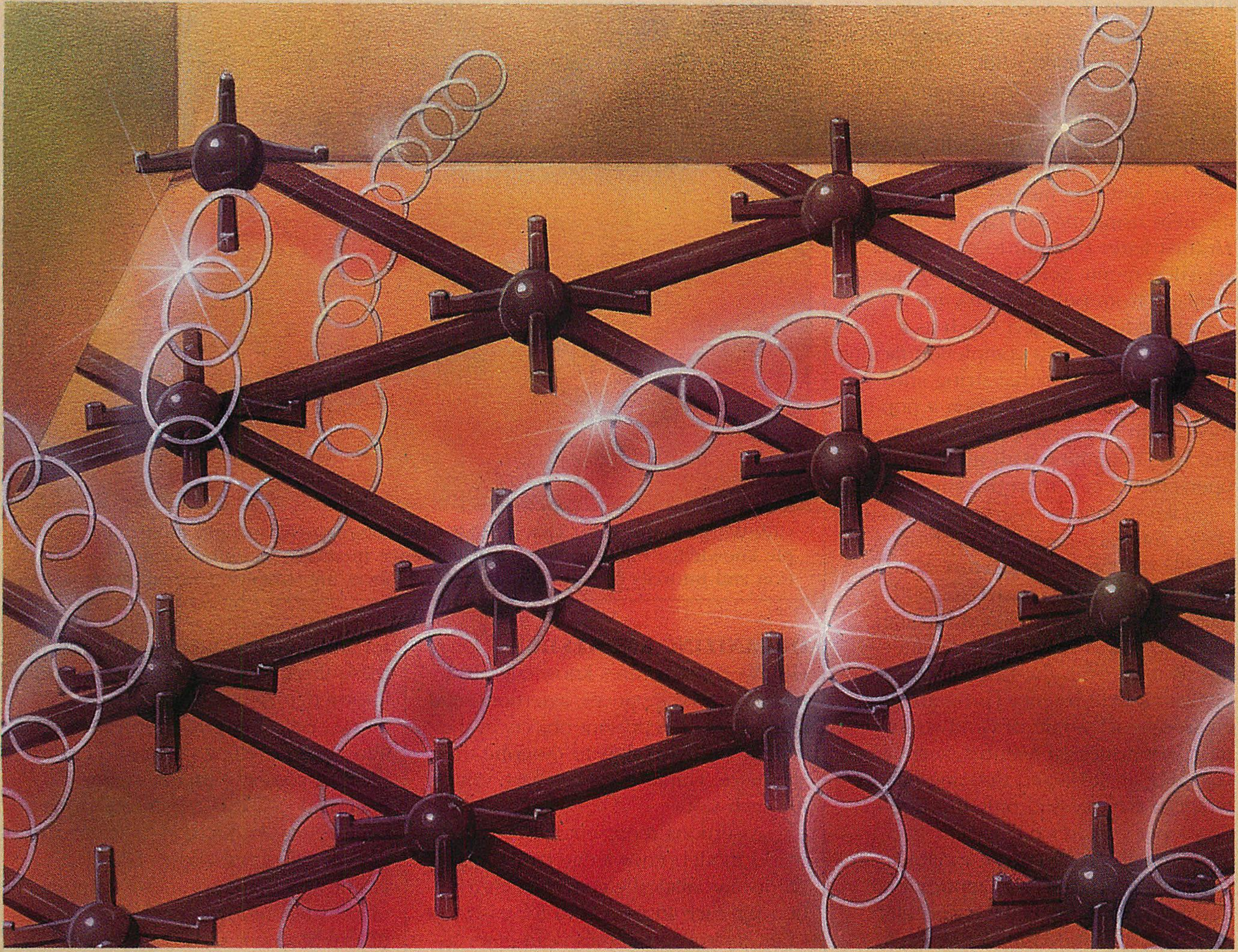


ILLUSTRATION • ANDY LEVINE

tems also require 5MB of extended memory (10MB is recommended). The hard-disk memory required is 10MB.

Gold Hill recommends an EGA display, but a monochrome monitor or CGA display can be used instead. A Microsoft or Mouse Systems mouse is recommended; a LOGITECH mouse used for this review had no problems. Performance is good on a 6MB Hummingboard but some features, such as garbage collection and GMACS tag tables, are faster with more memory.

The package consists of a set of 31 diskettes and 5 manuals, each in high-quality binders. The system diskettes take only 30 minutes to install; a log of all installation activities is kept. A separate diskette is provided to configure the Hummingboard.

INTERFACE INTRICACIES

The menu interface consists of a system of menus and windows that can be used by a nonprogrammer to define rules and frames, enter assertions, and run an application. The main menu consists of a menu bar across the top of the screen and a cursor-documentation line at the bottom. The menu bar contains the following items:

- System—provides access to GoldWorks utilities. The System pull-down menu includes Version, to show the GoldWorks configuration; Load and Save, to load files or save the current application in a set of designated files; Introduction, to display a customized start-up screen for an application; Tutorial, to invoke the GoldWorks tutorial; GMACS, to enter the editor; Top Level LISP, to enter the LISP programming environment in the developer's interface; DOS, to enter the DOS environment; and Exit, to leave GoldWorks (files will not be saved, so the user should use the Save selection before exiting).
- History—displays a list of the names of the last 20 items viewed using the menu interface's Browser and Inspector facilities. The Browser gets a picture of where an object is situated relative to other objects of its type, while the Inspector gets a detailed description of objects. The Browser facility in photo 1 shows a parent frame defined for the sample application (on the left) and child frames in a window on the right. The Inspector zooms in on the details of an object; in photo 2, the Inspector lists all assertions in the knowledge base.
- Define—enters the definition of a GoldWorks object from the menu interface. Objects include relations, frames, instances, sponsors, rules, rule sets, assertions, and attempts. Selecting one brings up a secondary menu used to interactively define that object. To assist users, developers can define objects in terms of previously defined objects.
- Find—enters the Inspector or the Browser for an item. Any GoldWorks object can be examined using this facility and can be brought into the Inspector by clicking on its name. If the display consists of more than one screen of items, the user can type the first letters of the item name and then press Enter. Like command completion, this displays the item or list of items that begin with these letters.

- Run—starts the current application.
- Debug—lists all events (breakpoints) that have occurred while running an application, defines breakpoints, and lists warning messages.
- Help—provides on-line help.

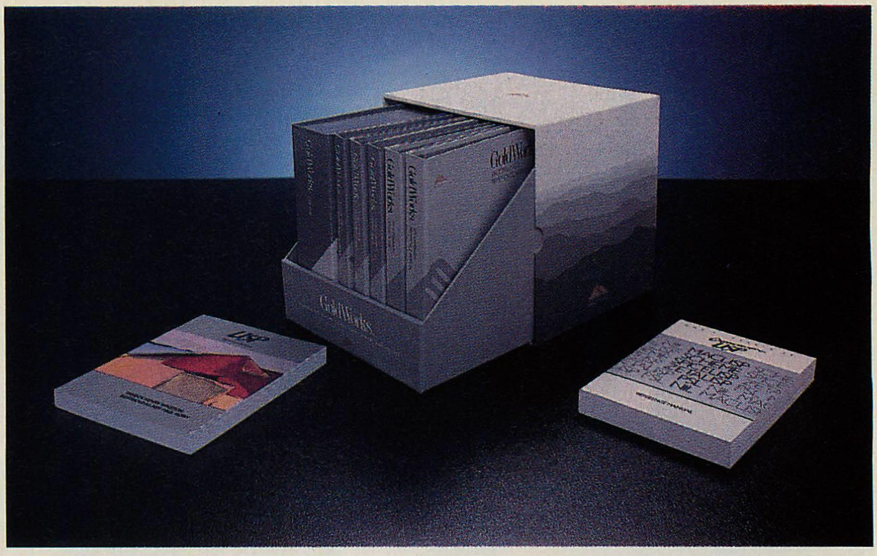
Users can choose these items by pointing with the mouse, cursoring to the item using the key pad, or entering Alt-(first letter of item) to pull down a selection menu. Selections are made similarly from pull-down menus. Searching for an object (such as a frame) is eased by a completion facility: the user enters a partial string and presses the spacebar to open a pop-up menu with a complete list of items of that type beginning with the string (or opens the item if there is only one).

Like the menu interface, the developer's interface offers many choices. GoldWorks' open architecture allows the screen toolkit, Lotus 1-2-3, and dBASE III PLUS to be accessed through this interface. The developer's interface is used from the GMACS editor or from a LISP listener (programming environment). Some of GoldWorks' more sophisticated features, such as changing agendas and defining handlers, can be accessed only through the developer's interface.

Predefined functions, variables, and macros are available for application development using GMACS and LISP. Macros are included to define each type of object.

The developer's interface allows incremental development of an application, which enables object debugging without having to evaluate an entire buffer or application. A rule can be defined and edited in GMACS, for example, and then immediately evaluated. The system returns any errors associated with the rule so that it can be edited and reevaluated. An application also can be run from GMACS. Text output from the application goes to a buffer and the user can then either continue the run or abort it. Alternatively, the LISP-interact mode of GMACS can be used to enter LISP and GoldWorks commands and interactively interpret them.

To save memory, an application can be built without the menu interface being present. If the menu interface is present, the developer can use it to define rules and other objects and then debug in GMACS, using the LISP-listener. The developer's interface provides functions to run an application (gw-run), view an object by name (gw-view), and put an object's definition into a GMACS buffer (gw-object).



GoldWorks is an expert system shell with comprehensive application tools, multiple knowledge representation, and inference methods. Its extensive documentation and tutorials educate and guide developers implementing expert systems.

ORGANIZING KNOWLEDGE

GoldWorks structures knowledge in frames, frame instances, assertions, and rules. Frames represent objects in the system and have slots to describe their attributes. Frame instances have values for each slot.

Assertions represent facts in the world model (knowledge base). GoldWorks automatically generates an assertion for each instance and for each slot of an instance; users also can enter assertions. See figure 2 for sample-application frames. The Purchasable-object- and Computer-object-frames are parents of the CPU-object-frame, so the latter inherits their slot definitions. When an instance of the CPU-object-frame is created, GoldWorks automatically makes an assertion that represents the instance itself and individual assertions for each slot value (see figure 3).

If . . . Then rules are used for inferencing in GoldWorks. When the if side of a rule (antecedent) matches an assertion pattern in the knowledge base, the rule fires; that is, the then side (consequent) is concluded and a new assertion might be placed in the knowledge base and then could be used to match other rules. For example, when the antecedent in the rule in figure 4 matches the assertion SYSTEM-SW CPU-11 DOS 2.0+, the new assertion, OS-TYPE CPU-11 DOS 2.0+, is added to the knowledge base.

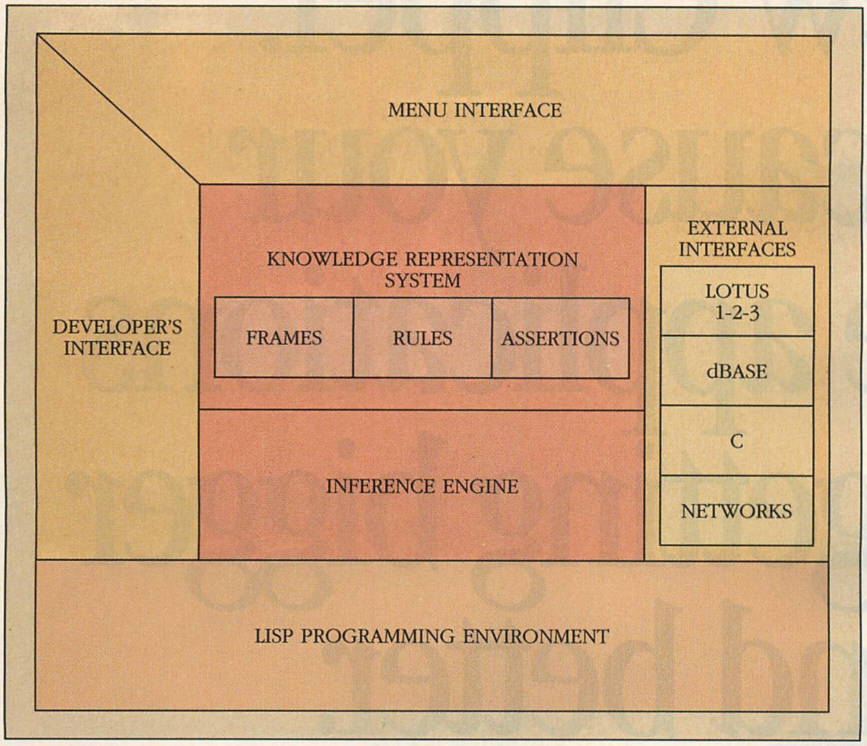
The knowledge base in GoldWorks is memory-resident and can be saved on disk for world loads. Both the developer and menu interfaces include tools to enter, modify, and delete frames, assertions, and rules. The careful and controlled interplay between these three objects is the foundation of knowledge representation in GoldWorks. Each has powerful features that allow the developer to create a versatile and complex expert system.

Frames. GoldWorks' frames provide a comprehensive method of structuring knowledge about classes of objects. Frames can have an unlimited number of slots to hold information about an object. Slots are further defined by nine types of GoldWorks-supported facets: Print name (to assign a print name to the slot), Documentation string (to enter a string of text that describes the slot), Explanation string (to add text to the end of the system-generated explanation for slot-value assertions associated with the slot), Constraint (to specify limitations on the values or types of values a slot can have), Multivalued (to indicate whether the slot can accept more than one value), Default value (to provide values when none are entered), Certainty (to provide a certainty level), When-modified (to be executed when a slot value is added or modified), and User (to hold a user-defined LISP form).

Supported constraints include: One-of (list of allowable values), Range (minimum and maximum values), LISP-type (list of allowable Common LISP or user-defined data types), Instance-of (name of an instance the slot will inherit from), and Child-frame-of (name of the parent frame for the frame that is the value of this slot).

Inheritance of slot values and facets is controlled in GoldWorks by structuring frames into a lattice. The

FIGURE 1: ***GoldWorks' Architecture***



GoldWorks is a comprehensive expert system shell with a layered design. The top layer consists of a menu interface, developer's interface, and interfaces to other programs; the middle layer consists of the knowledge representation and inferencing methods; the lowest layer is the Golden Common LISP environment.

lattice structure allows multiple inheritance (see figure 2), which is inheritance from more than one parent.

Frames are placed in the lattice and child-parent links established when the frame is defined using either the Define macro in the developer's interface or the Define: Frame menu selection. When using the menu, the developer is prompted for the name of a parent frame. If no parent is specified, the frame is attached to the GoldWorks root frame, Top-frame, to insure that the lattice is always rooted. The Frame Inspector screen provides the Parent Frames and Child Frames menus to allow the developer to view, position, or disconnect parent and child frames.

The menu interface is aware of the semantics of the lattice: if frame *x* is the parent of frame *y*, and frame *y* is the parent of frames *v* and *w*, then removing frame *y* will automatically assign frames *v* and *w* to be children of frame *x*. The menu interface assists the developer in keeping the lattice consistent. GoldWorks will generate an error message if adding or disconnecting a frame causes conflicts within the lattice (for example, adding a frame as a child to another frame when it is already the parent).

When a frame inherits slots sharing the same name from multiple parents, one of four methods is used to define the facets: shadowing, overriding, merging, or intersecting. In shadowing, a local value given to the facet will replace the inherited value, otherwise the facet is inherited from the parent that was defined first. A yes in the Multivalued facet overrides a no. When-modified and User facets merge when inherited from multiple parents; that is, the functions are combined into one list that is arranged in the order in which the parents were defined. Constraint facets, of the type One-of and Range, are merged by intersection from multiple parents. Deleting items that are in the lattice automatically causes the definition of subordinate frames in the lattice to be reevaluated.

The When-modified facet is for attaching a list of LISP functions (demons) to be called whenever a slot value is asserted, retracted, or modified. This is called access-oriented or event-driven programming. A demon function is given the instance and slot names and old and new values and is called once for each value of a multivalued slot. User facets hold additional user-dependent slot values, including any LISP form or structure (such as a specification list). For example, a display program can have demons on the slots of frames that represent objects to be displayed on the screen. The demon function could be used to ask an object to redraw itself when it is modified.

Assertions. GoldWorks supports structured assertions (which are entered by GoldWorks into the assertion base automatically when frame instances and instance-slot values are defined) and unstructured assertions (entered by the developer from the menu or developer's interface and having no associated frame or slot value).

Unstructured assertions can be one of three types: assertions, relational assertions, or functional assertions. Regular assertions are general facts about the knowledge domain. Relational and functional assertions result from defining relations using either the define-relation macro in the developer's interface or the define-relations option from the menu interface. Relational assertions allow developers to enter a set of many assertions based on the same pattern. For example, the following relation can be entered:

```
(define-relation cable
(:relation-type: assertion)
(insulation wire-type number-conductors))
```

The relation cable is the basis for the following assertions:

```
(cable teflon copper 9)
(cable plastic goldplated-copper 25)
```

GoldWorks provides functional assertion relations to prevent the developer or user from entering conflicting assertions in the assertion base:

```
(define-relation monitor
(:assertion-type: functional assertion)
(monitor customer pixels))
```

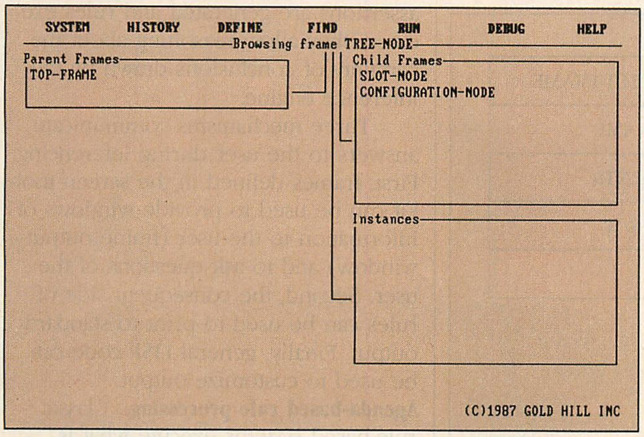
If two functional assertions are entered, both identical in all but the last value, the first will be retracted when the second is entered. For example, assuming a customer is planning to purchase a monitor, if the following assertions are entered in this order,

```
(monitor customerA 640x480)
(monitor customerA 960x720)
```

then the 640x480 assertion is retracted. The developer uses this powerful feature to prevent contradictory assertions from being active in the same world model during a single session with the expert system.

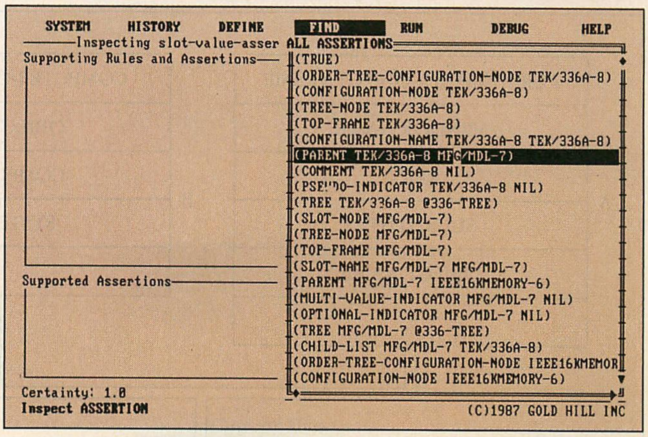
Support of unstructured assertions by GoldWorks not only provides the developer with the versatility of adding

PHOTO 1: ***Browsing Facility***



The Browser is used to examine the relation of any object (frames, instances, and sponsors) to other similar objects in the system. Here, the root frame, Top-frame, has two child frames, Slot-node and Configuration-node.

PHOTO 2: ***Inspecting Facility***



The Inspector is used to zoom in on the details of any object in the knowledge base; for example, inspecting an instance reveals all slot values. In this case, all assertions in the knowledge base of the sample application are examined.

assertions without defining frames, but also allows a conventional program running in a GoldWorks environment to enter, directly into the knowledge base, an assertion containing a pattern that the driving software needs to trigger matching rules to fire.

GoldWorks' assertions have three types of information: the derivation (how the assertion was put into the assertion base), one or more justifications (why the assertion is currently in the assertion base), and a certainty factor. The dependency information is important for maintaining the integrity of the knowledge base by controlling the retraction (deletion) of assertions from the assertion base.

When a frame instance is deleted from the lattice, all structured assertions resulting from instantiation of that frame also must be retracted. When an assertion is retracted, the system maintains consistency by using justifications to identify and retract all dependent assertions. If all justifications for an assertion are retracted, the assertion also is retracted. The system displays all dependent assertions to alert the user to the scope of a retraction. The dependency network can be examined by selecting the Assertions item from the Relation Inspector screen.

The Explain function traces through an assertion's dependencies and builds an explanation string for current derivation of an assertion. If any assertions in the derivation are retracted, the explanation shows only the remaining derivation. Explain is useful for debugging the system and building user-oriented explanations of the state of the knowledge base. The developer can use it to structure explanations to the user. Although the explanation string is sparse for a total explanation, it provides core information needed to write a more complete explanation.

Finally, LISP-function assertion relations are used as predicates in the antecedents of rules. Some predicates are predefined, such as the following: equal, not equal, = (equal for numbers), /= (not equal for numbers), <, <=, >, and >=.

Rules. GoldWorks supports forward rules (for forward chaining and, in rule sets, for goal-directed forward chaining), backward rules (used in backward chaining), and bidirectional rules (used in both). Rules are used during inferencing to match against assertions in the knowledge base. Rules have a print name, documentation string, explanation string, direction (forward, backward, or bidirectional), sponsor and priority (used in conflict resolution), certainty factor, and dependency value.

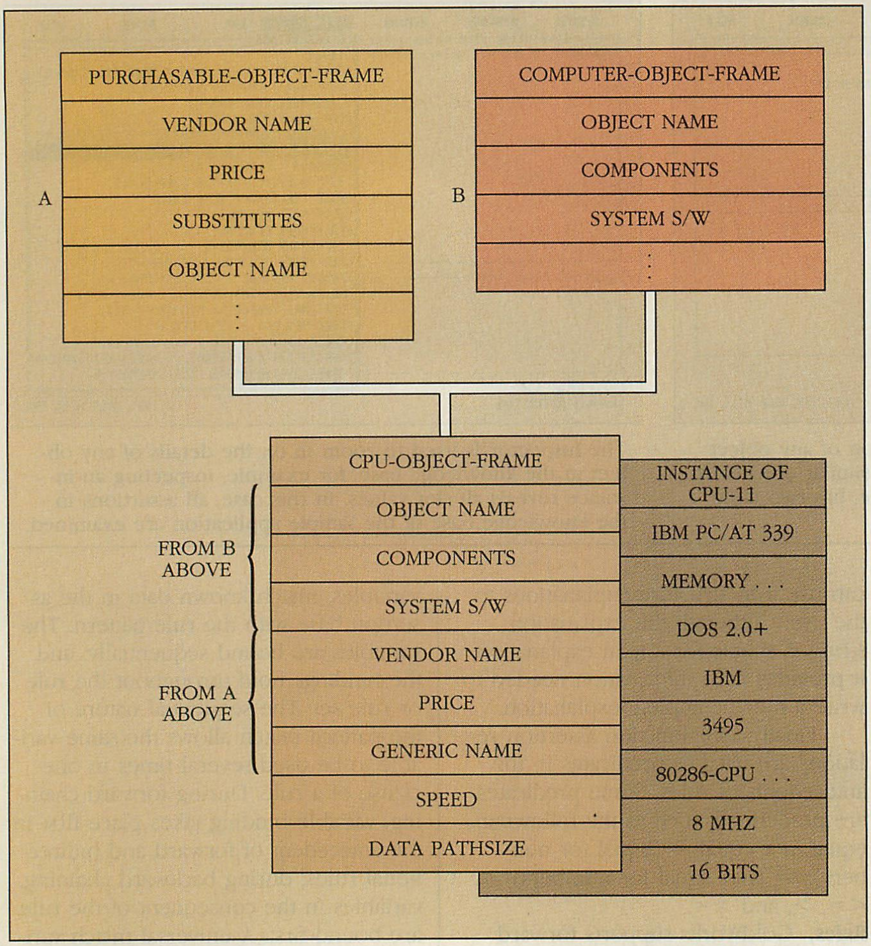
Setting the dependency value to *true* tells the system to create a justification (why the rule is in the knowledge base) when the rule fires. If assertions that match a rule (causing it to fire) are retracted, the assertions that resulted from firing that rule also will be retracted. The dependency value is set to *nil* for a state rule (one that describes a change of state), which does not support the retraction of dependent assertions. The consequent of the rule can have an and-then clause that will be invoked after the rule is run.

Variables can be included in rules by preceding the variable name with a ?. The system tries to find a set of values and objects that, when bound to variables, match known data in the assertion base with the rule pattern. The variables are bound sequentially, and the bindings hold throughout the rule or rule set. The sequential nature of the pattern match allows the same variable to be used several times in one clause of a rule. During forward chaining, variable binding takes place first in the antecedent of forward and bidirectional rules; during backward chaining, variables in the consequent of the rule are bound first. A universal match variable, ?, is included that is not bound and can be used more than once to refer to different objects.

The antecedent of rules can match on variables, assertions, and instances of frames (with or without restrictions on specific slot values). Assertions also can be bound to variables, and later retracted in the consequent of rules. The consequent side of rules can be used to make and retract assertions, delete instances, print out text to the standard-output stream, and evaluate general LISP expressions.

When the define-rule option is selected from the menu, a submenu bar appears with templates, commands, parameters, enter, cancel, and help options. If . . . Then key words are displayed in an editing area; the user can enter the text for the rule directly or use the menu items at the top of the screen to help format the rule. A parenthesis checker places a rectangle over the open parenthesis that corresponds to any closed parenthesis entered. These aids help the less experienced developer to enter rules; they can be ignored when it is quicker to enter rules directly.

FIGURE 2: ***Sample Frames Showing Lattice Structure***



CPU-object-frame describes a specific CPU in the computer configuration application. It inherits slots from its parents, Purchasable-object-frame and Computer-object-frame. In addition, the new slots, speed, and data-path size are defined. CPU-11, is an instance of the CPU-object-frame and has actual slot values.

Lattice objects and relations can be included in the rule by selecting the templates option from the menu bar. The system displays a form that provides structured help for entering lattice objects into rule form, including fields for the frame name, the instance name, and any slot names and values required. The commands option is used to enter GMACS (to edit the rule), to view the LISP format of the rule (read-only), to view errors currently associated with the rule, to pretty-print (nicely indent) the rule, to delete the rule, to set a breakpoint at the rule, or to enter DOS. The parameters option is used to enter the print name of the rule, a documentation line, the rule direction, a priority, explanation string, dependency status, sponsor, and certainty factor. Selecting the enter option runs a rule validation checker.

GoldWorks allows rules to be collected into sets for use by the system during inferencing. The set has a pattern, called the enabling pattern, that must be matched for the system to use the individual rules.

INFERENCING APLENTY

The GoldWorks system includes a variety of inferencing mechanisms, such as agenda-based rule processing, forward and backward chaining, goal-directed forward chaining, and object programming, that gives the user a tremendous amount of flexibility in building expert systems. Users new to expert systems development at first might choose to use only the most basic capabilities such as forward and backward chaining, adding more sophisticated features as needed.

GoldWorks uses a Rete network to minimize the time required to match the assertions to the rule base. This network keeps track of partial matches to rule clauses and binding of variables to patterns in the clauses, and insures that rules fire only once for each set of assertions they match. Certainty factors also can be attached to rules. When assertions are generated and rules executed, they allow reasoning about the certainty of conclusions drawn by the inference engine.

Three mechanisms communicate answers to the user during inferencing. First, frames defined in the screen toolkit can be used to provide windows of information to the user (frame output-window) and to ask questions of the user. Second, the consequent side of rules can be used to print to standard output. Finally, general LISP code can be used to customize output.

Agenda-based rule processing. Classic rule-based systems execute what is known as the recognize-act cycle. All rules whose antecedents match assertions in the knowledge base are found, then one is chosen to fire. In any cycle, more than one rule is able to fire; choosing which of many possible matching rules to fire is known as conflict resolution. To control conflict resolution, GoldWorks uses sponsors and agendas. A sponsor is a prioritized list of rules waiting to fire and a state code to indicate if the sponsor is enabled (active) or disabled (inactive).

When a rule is matched, it is put on a sponsor as an agenda item. Forward agenda items are created by GoldWorks from forward and bidirectional rules, the set of assertions that match the antecedent pattern of the rule, and the bindings for the match. Backward agenda items consist of a rule whose consequent pattern matches the goal pattern of a goal (attempt) and the bindings for the match.

Sponsors are defined using the Define-: Sponsor option on the main menu or the Define Sponsor macro in the developer's interface. The developer specifies how agenda items having the same priority are to be added to the sponsor: depth-first (last in, first out) is the default for backward agenda items; breadth-first (first in, first out) is the default for forward agenda items. In the GoldWorks system, sponsors are organized in a tree (see figure 5) and can be placed anywhere in the hierarchy, with the GoldWorks default, Top-Sponsor, as the root.

Developers can assign agenda items to specific sponsors. This powerful feature of GoldWorks allows advanced developers to control the flow of inferencing in large systems.

Sponsors can be added, deleted, enabled, and disabled. Disabling or deleting a sponsor affects all its subordinates. Enabling and disabling spon-

sors can be used to control the order of firing of rules on agendas. Functions also are provided to access the list of items on a sponsor's agenda, change the order of agenda items, or delete items. Individual agenda items also can be fired, independent of the sponsor-traversal algorithm.

Inferencing is done in depth-first passes through the sponsor hierarchy. Users can set the default-quanta variable to specify the number of agenda items to be fired during one pass. This value is divided equally between the top sponsor and its immediate child sponsors in the tree. Thus, as many agenda items as possible are fired from top sponsor, up to the value of its allocation; any remaining quanta are passed down to the first subsponsor during the depth-first traversal.

This continues until all sponsors are processed. Firing agenda items typically creates new items; the system continues the depth-first traversal and begins the cycle again with a new quanta until no more agenda items remain on the agenda of any sponsor.

This scheduling algorithm is designed to guarantee the running of agenda items on sponsors near the top of the hierarchy, with as many low-priority agenda items as possible running within the allocated quanta of rule firings. If all agenda items must be run for all sponsors during a single pass of the agenda hierarchy, the default-quanta must be set high.

The combination of prioritized rules with sponsors and agendas is used in GoldWorks to provide an event-driven mechanism for orderly running multiple sets of rules. This is unique to GoldWorks and is an improvement over traditional rule-based inference engines that implicitly use a single agenda of prioritized rules. For example, *meta-rules* (rules that are used to control the order of firing regular rules) can be put on a separate, high-priority sponsor. Meta-rules would then be run first by GoldWorks, and could be used to determine the order of firing other rules.

An example of agenda-based processing using meta-rules and sponsors in the sample application is shown in figure 5. Sponsor 1 contains agenda items for meta-rules that determine the order of processing other rules by rearranging agenda items on the other sponsors. Sponsor 2 contains agenda items that result from constraint violations in configuring computers. These items are generated when matched components are unworkable. Sponsor 3 represents equipment found in inventory. The order of firing sponsors (1, 2, 3) forces only valid configurations.

The sponsor hierarchy can add flexibility to applications requiring it. Those not requiring it can ignore the agenda structure, resulting in classic forward or backward chaining. The new developer can work at a high level and ignore the subtleties of the sponsor hierarchy, while the experienced designer is given facilities for designing scheduling strategies.

Forward chaining. Forward chaining is used to infer all possible solutions from a given set of assertions. All rules whose antecedent pattern matches the assertion base with their bound variables are used to create agenda items. These matches are placed on the agenda of a given sponsor in priority order: depth-first or breadth-first. After all matches are done, the highest-priority item is removed from the agenda and fired; this can cause more matches and subsequent agenda items. This continues until no more rules are found whose antecedents match assertions in the assertion base, or until a solution is found.

Backward chaining. Backward chaining directs the search toward a particular goal. It is initiated by a user creating an attempt and querying it. An attempt consists of a pattern corresponding to the goal (the pattern to be matched), a state (to keep track of how far the search for the pattern has gone), and a sponsor (to hold the agenda items resulting from successful matches).

Four types of queries for backward chaining can be made: query, query-all, query-initial, and query-once. Query is used to find one match. It keeps track of the last match found, for further queries. Query-all is used to find all matches for the goal pattern. Query-initial finds one match and resets the state so that the next query will begin again with the first match. Query-once is used when only one query against a goal pattern is made; it responds with the same data as query-initial but does not create an attempt structure.

GoldWorks implements backward chaining by searching for assertions that match the attempt (goal) pattern; if these are found, they are returned and backward chaining concludes. If they are not found, the system searches for backward or bidirectional rules whose consequent pattern matches the attempt pattern. All matching rules are put on the attempt's agenda (by priority, breadth-first within priority).

The system creates a sub-attempt (sub-goal) out of the antecedent of this rule, looks for an assertion to match it, and if that fails, for a rule consequent to match it. The process continues with further sub-attempts created and processed depth first; backtracking occurs as necessary when sub-attempts fail at any level. Once the system matches a sub-attempt with an assertion in the assertion base, it forward chains through rules on the agenda until the original attempt is satisfied. The and-then portion of the rule will be fired, usually printing the query answer.

Goal-directed forward chaining. Goal-directed forward chaining integrates forward and backward chaining. It uses rule sets to group forward rules together with a specific enabling pattern. When the pattern matches the goal pattern of the attempt queried, the individual rules in the set are eligible for

FIGURE 3: *Assertions*

```
FROM      ( INSTANCE  CPU-11 IS CPU-OBJECT
INSTANCE        WITH  OBJECT-NAME IBM PC AT 339
                WITH  COMPONENTS (MEMORY,...)
                WITH  SYSTEM-SOFTWARE DOS 2.0 +
                WITH  VENDOR-NAME IBM
                WITH  PRICE $3495
                WITH  GENERIC NAME (80286,...)
                WITH  SPEED 8MHZ
                WITH DATA-PATH-SIZE 16 BITS)

FROM      ( CPU-OBJECT  CPU-11)
SLOTS     ( OBJECT-NAME CPU-11 IBM PC AT 339)
          ( COMPONENTS CPU-11 (MEMORY,...))
          ( SYSTEM-SOFTWARE CPU-11 DOS 2.0 +
          ( VENDOR-NAME CPU-11 IBM)
          ( PRICE CPU-11 $3495)
          ( GENERIC NAME (80286,...))
          ( SPEED CPU-11 8MHZ)
          ( DATA-PATH-SIZE CPU-11 16 BITS)
```

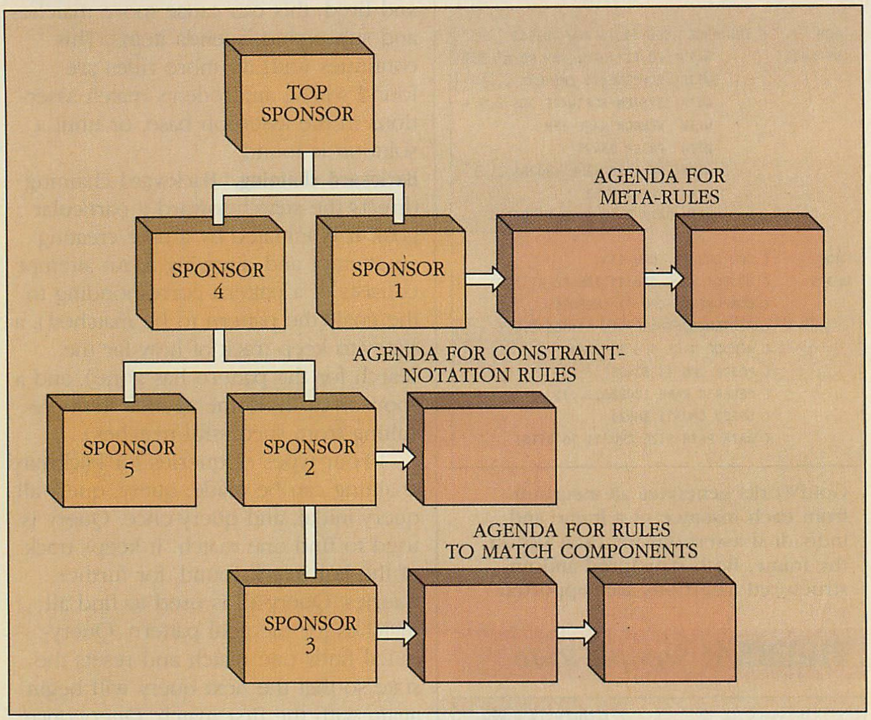
GoldWorks generates an assertion from each instance of a frame and individual assertions for each slot in the frame. Both structured and unstructured assertions are supported.

FIGURE 4: *Sample Rule*

```
IF   (INSTANCE ?X IS            ;if there is a CPU
     CPU-OBJECT
     WITH SYSTEM-SOFTWARE ?S ;with system software

     WITH OS-TYPE NIL)          ;but no operating
                                ;system type
THEN (INSTANCE ?X IS            ;then set the
     CPU-OBJECT
     WITH OS-TYPE ?S)           ;OS-TYPE to SYSTEM-
                                ;SOFTWARE

NEW ASSERTION GENERATED:
(OS-TYPE CPU-11 DOS 2.0 +)
```

A rule consists of antecedent and consequent clauses. Variables are designated by preceding a name with a ?. When a rule matches an assertion, a new assertion can be generated.

FIGURE 5: ***Sponsor Hierarchy to Control Rule Firing***



Sponsors are prioritized lists of agenda items waiting to fire. The agenda is ordered in a tree with those at the top having the greatest chance of firing during inferencing. Here, Sponsor 1 holds a list of meta-rules (rules about rules) that control firing of items on Sponsor 2; thus, developers can help schedule strategies.

forward chaining. A rule set enables the system to ignore the rules in the set when they do not fit the current situation (when other attempts are running). Only the rule set, not each rule in the set, is known to the system until the set is enabled.

The system creates an agenda item out of the rule-set pattern and the binding of the variables in the enabling pattern. When a rule set agenda item fires, the system creates forward agenda items out of forward rules in the set that are ready to fire, and places them on the agenda by priority. Bindings from the enabling pattern apply to all rules in a set. If a rule set is given a high priority, the forward rules on the rule set can be run before any backward-chaining rules (in order to obtain inferences needed by the backward-chaining rules). If a rule set is given a low priority, it runs after all backward chaining rules (perhaps to ask the user to input a value if backward chaining fails to provide it).

Figure 6 shows a rule set that has the goal of finding the correct CPU type for a user requiring WordPerfect. The enabling pattern "choosing CPU type" is used to hide these rules from the inference engine, except when a CPU is selected. Rules can be generated during the process of generating rules for configuring systems including WordPerfect, with knowledge factored away to be used as required. Similar rule sets could be generated for each product in the configuration.

Object programming. Here, data structures are treated as active objects that can receive messages to make changes. A display window on a screen can be sent a message to change the cursor position, output a character string at the present cursor position, or scale the display of its own image.

Object programming in GoldWorks is done by attaching LISP functions (handlers) to frame objects and sending messages to instances of the frame to which they are attached. Handlers are defined using the developer's interface and are inherited in the frame lattice. The function **send-msg** is used to invoke a handler at the instance level. This function can be used in rules.

Handlers can send messages to other instances to invoke other handlers, thereby implementing a message-passing style of programming. The function **instance-all-handlers** returns a list of all the handler names defined for a given instance.

The sample application might use demons and handlers to enforce the following constraint: "If software includes WordPerfect 4.2, the operating system cannot be DOS 1.0." This can be implemented by defining a demon to be run when the Version slot in the frame for DOS is set, and another to be run when the Version slot in the frame for WordPerfect is set. The first demon sends a message to invoke a handler that checks to see if the version of DOS is incompatible with any software requested. The second sends a message to invoke a handler that checks to see if the version of DOS is less than 2.0. The actual test to be performed also could be implemented by a handler that checks a constraint frame that defines the constraint. GoldWorks has predefined slot-accessor functions to modify a slot value or to add or retract assertions.

Certainty factors. GoldWorks hedges in its use of certainty factors: a simple scheme is provided with the system, along with the hooks required to support more complex user-written schemes. The default scheme uses a numerical scale from 0.0 to 1.0; developers assign values subjectively on the validity of an assertion.

When assertions are made as a rule fires, the system combines the certainties of the assertions that match the antecedent pattern of the rule with the certainty factor of the rule itself to derive a certainty factor of the new assertion. Assertions entered into the knowledge base as a result of a default (slot) value are assigned a default (slot) certainty (or 1.0, if the slot certainty is not filled in). Global variables define a user-defined certainty function for rule firings and set the default certainty of an assertion. The certainty function also can be set to *nil* to disable certainty function calculation.

The default certainty function takes the minimum value for all the antecedent assertion certainties and multiplies it by the certainty of the rule. If an assertion is in the knowledge base and made again, the certainty is updated according to the formula:

$$\langle\text{result-certainty}\rangle = \langle\text{new-certainty}\rangle + \langle\text{old-certainty}\rangle * (1.0 - \langle\text{new-certainty}\rangle).$$

DEVELOPER HELPERS

The GoldWorks screen toolkit is used to create temporary menus (to query or notify the user), output windows (to display text to the user), and permanent screens and menus (to customize a menu interface for an application). These can be used by the developer as well as by GoldWorks itself to communicate with the user.

Menus for user input can be permanent or pop-up. Pop-up menus include popup-confirm (asks the user a yes-or-no question), popup-ask-user (asks the user to enter a value), and popup-choose (asks the user to choose one of several strings). Output windows display information such as text and warnings; they are permanent (not pop-up), cannot be exposed or hidden, and do not accept user input. Menus and windows are integrated into screens, and include a menu bar with attached pull-down windows.

The developer uses the toolkit's predefined frames (such as popup-ask-user, screen-layout, and output-window) to create the menus, windows, and screens. The developer defines an instance of one of these frames and a rule that will cause the menu to appear. Pop-up menus have a go slot; when a rule or other function writes yes into the go slot, the menu is activated. The slot value is retracted to make it available again.

GoldWorks applications can directly access files produced by dBASE III PLUS either through LISP functions or by making instances of an interface frame. The interface frame can be used to read and write records and fields; open, access and create index files; and read and write memo files.

The interface to Lotus 1-2-3 is similarly implemented through LISP functions or an interface frame, called 123-action. This frame has slots for the spreadsheet name, the action desired, start and end row and column, the written or returned value, error code and message, and, similar to the screen toolkit, a go slot. The Lotus 1-2-3 interface supports making, loading, and saving spreadsheets; reading and writing values, formulas, and formats (and their ranges); and getting and setting wrap values and orientation.

An initialization file is used for configuring the system with or without various facilities, including the menu interface, the screen toolkit, and the Lotus 1-2-3 and dBASE III PLUS interfaces. The Lotus 1-2-3 and dBASE III PLUS interfaces also can be loaded or unloaded under program control. They are easy to use and well documented. Other ones, such as an SQL interface, would be useful additions.

Debugging capabilities, including Breakpoint, Events, and Warnings, are available by selecting Debug from the main menu bar. Breakpoints can be set on sponsors, rules, agenda items, or assertions to stop the inference engine at specified points during execution of an application. They can be set to occur when items are put on the sponsor's agenda, when a rule is about to fire, when a fact is asserted, and so on. The developer can select from two actions at breakpoints: Trace causes the system to print the name of the breakpoint object on the Events menu without stopping the application, and Stop halts processing.

Events lists all breakpoints occurring during an application run. When the system encounters a breakpoint for which the Stop option has been selected, the Events menu list automatically appears on the screen. When a breakpoint occurs during or after running an application, selecting Events shows all breakpoints encountered during the run. Any object on the menu can be examined using the Browser or Inspector. Pressing Enter will resume the application.

With the Warning option, the developer can list all GoldWorks warnings encountered while creating or running an application. Selecting one of these warnings invokes the Inspector for the object that caused the warning.

GoldWorks' comprehensive on-line help facility is available for all commands at the menu interface, including cross referencing and topic words that can be selected for easy access to related information. Each help screen includes highlighted words; selecting one of these gives a text screen explaining the topic. Help-related menu commands include Help (to provide introductory information about the help facility), Search (to invoke a menu enabling search for specific topics), Index (to get a list of the major topics covered from which selections can be made), Backtrack (to move to the previous help screen), and Quit. Some menu screens include the word *menu* on the command line; selecting this displays a menu of topics related to the current help screen.

An extensive on-line tutorial to using GoldWorks includes examples on the use of each feature. A reference manual explains the theoretical foundations of GoldWorks and information useful for structuring the knowledge base, including defining frames, slots, and facets, and writing rules and rule sets. A user's guide gives in-depth information on the menu interface and presents example expert systems. The quality of the documentation is good, but it could use a better introduction to building expert systems and using GoldWorks for this purpose, although the tutorial fills some of this need.

FIGURE 6: ***Rule Set***

```
(DEFINE-RULE-SET  WD-PERFECT-OS-RULES
  (doc-string  "Contains forward rules for
                choosing operating system when
                software includes WordPerfect")

  (choose-os  ?CPU  ?OS-TYPE)  ;enabling pattern
    .
    .
    .
  (DEFINE RULE PC-RULE()
  (INSTANCE  ?X IS CPU-OBJECT
       WITH  GENERIC-NAME 80286-CPU
       WITH  OS-TYPE  ?O)
  (COMPARE-OS-TYPES ?O DOS 2.0) ;function to test
                                ;for illegal value
  THEN                          ;then set up new
                                ;assertion
  ((CONSTRAINT-FAILURE OS-MISMATCH
                               WORD-PERFECT ?O))

      .
      .
      .
```

GoldWorks provides rule sets to allow the developer to factor knowledge into related groups. The system ignores the rules in the set when the enabling pattern is not matched.

THE SAMPLE APPLICATION

To observe the actual operation of GoldWorks in a problem-solving situation, a sample application called Expert Configurer was developed and tested. GoldWorks proved a powerful development tool.

In a retail computer store, customers might have a general idea of what they want (a color monitor, some memory, a disk, and a printer), but no specifics (a 286-based system with a 40MB disk, 2MB of extended memory, a 2,400-bps modem, a CGA, and color monitor). A salesperson who is expert in this field refines the original request to a more specific set of components and, knowing that speed is vital to a sale, makes recommendations based on stock on hand. If some components are not available, the expert substitutes others and knows what effects these have on the system.

The Expert Configurer is a tool to be used in lieu of a human expert to configure computer systems. It assists less-experienced salespersons in configuring systems without wasting time leafing through specification manuals. Such a program has to reason symbolically about the domain, its objects, and all subcomponents. It might include filling required components that are only implicit in the request (such as a serial port and display adapter).

The system must abide by rules or restrictions governing computer config-

urations. For a personal computer, rules address issues such as the power and number of bus slots available, the components requiring other components (for example, a modem requires a serial adapter), and the need to avoid unworkable combinations of components. Some rules are used to determine the order of matching requirements and others to decide how to handle partial matches, such as finding a 286 with everything required that has an extra disk drive but not a modem.

GoldWorks is an excellent choice for this application because it combines frames and rules whose clauses can match frame instance patterns, its rule sets limit searches during inferencing, and its developer's interface and screen toolkit can customize menu interfaces.

Modeling the domain. The Expert Configurer models each configuration as a hierarchy (tree) of objects, including constraints on possible combinations. The system uses configuration trees (all possible configurations for the system), inventory trees (actual equipment in stock), and order trees (all customer-acceptable configurations).

Each configuration tree represents a set of possible computer configuration. Child nodes represent components and subcomponents needed for the system, and their constraints and specifications. Each node includes a name, documentation text, and parent and child relationships. A parent node might have a property called bus slots or power. Each leaf node supplies a value for bus slots or power; constraints on the relationship of these values insure validity of the order.

For a 286-based computer, constraints include:

If color monitor, then color adapter
Number of AT bus slots must be greater or equal to number used

Specifications of the configuration might include:

Total number of AT bus slots = 8
Total power available = 175 watts

Specifications of a subconfiguration for a video controller might include:

Number of AT slots required = 1
Power required = 15 watts

To produce the order tree, the Expert Configurer queries the customer for all options he wants, while enforcing the constraints. It attempts to match the order tree to an inventory tree.

Building the Expert Configurer. The Expert Configurer was implemented as a combination of GoldWorks frames (representing each configuration and its components), rules and rule sets (representing the constraints and requirements for matching stock), and LISP code (for the menu interface and SQL-database interface).

The GoldWorks menu interface was used to define a small number of frames that defined items such as configurations, specifications, parent-child relationships, and so on. The definitions were captured in a file, to be loaded at runtime. Using the menu interface made the process easy—it was done only once for all configurations.

The menu interface was then used to build several prototypical configurations, each consisting of a number of

The GoldWorks developer's interface can be used to customize a menu interface for refining the configuration tree into an order tree.

frame instances. The developer's interface then was used to build tools that would allow people knowledgeable about the structure of the computer configurations, but not educated in GoldWorks, to enter the information. The developer's interface was used to translate this information into GoldWorks frame structures.

To store the configuration trees in an SQL database, indexed by the name of the configuration, an SQL interface was built using Golden Common LISP. The developer's interface was used to build a tool to read configuration trees from the database, turning them into frame instances and parent-child links between instances.

The time invested in building these tools was worth it because each configuration structure did not have to be defined individually. The tools also allow end users to enter information without being aware that they are building frame structures. The developer's interface combined with the screen toolkit makes special-purpose menu interfaces easy to build, and seamlessly integrates them with the rest of the GoldWorks-based application.

The menu interface Rule Editor was used to define GoldWorks rule sets, which are used to organize rules into small, well-focused groups. The state of the order-filling process determines which rule sets are available. Rule sets are defined to make strategy decisions concerning the trade-off between choosing final assemblies, subassemblies, and component parts, as well as for using stock in different states (for example, first try a final assembly that is on order, or first try subassemblies that are available).

The developer's interface can be used to customize a menu interface for refining the configuration tree into an order tree. The user can make choices implicit in configuration hierarchy (such as monochrome, Hercules, CGA, EGA, or VGA). The system follows child links in frame structures and poses questions to the user, checking the developing order tree as the questions are answered to allow only valid combinations. Experiments were done using When-added facets (demons) and rule sets to insure valid combinations. When-added facets have the advantage of being localized to the frame instance in question, but they must be put on all frame instances that are involved in a given constraint. Rule sets can be kept small and enabled by the frame instances in question.

The next step is to select stock by matching order trees to inventory trees. If the Expert Configurer cannot pull the exact configuration desired from inventory trees, it must decide whether to pull an already-configured substitute or build a configuration from separate components. If a partial match is found, the name of the missing component is returned. A database is queried to get the names of all nodes containing substitute or missing components, and the matching process continues.

Many benefits of expert systems become obvious at this point. The same frame base used to run this process, for example, can be used to define a facility that explains the process to the user. Changes to configuration hierarchies also were easy with tools built with the developer's interface, and changes to processing strategies were eased by manipulating rule sets.

Developing such a system involves much time spent tweaking rules and rule sets. In the sample application, the GoldWorks debugging capabilities were paramount—the ability to breakpoint on a rule being put on an agenda or on a rule firing can be used to monitor the configuration process, along with incremental changes to the rule base.

As processing strategies grew more and more complex, the developers chose to write rules that considered

IBM®

Who says you have to pay extra to get an ASCII terminal equipped with these three letters?

Introducing the new family of IBM 3151 ASCII displays.

Now you can have the most versatile ASCII terminals IBM has ever made, for the lowest price IBM has ever offered. Our new 3151 family gives you more functions, and greater compatibility with more ASCII host system computers, for single unit purchase prices starting at less than $400* per terminal.

Three models and up to 16 emulations make them flexible.

The entry level Model 110 comes with 10 non-IBM emulations built in, and provides an 84-key keyboard with 12 definable function keys.

The full-function Models 310 and 410 come with 11 emulations, and are easily capable of more (such as DEC VT220/100/52™ and WYSE WY-50/50 +™) by simply adding a new low-cost, slimline cartridge.

Their 102-key keyboards, equipped with up to 36 definable function keys, are also recappable, so you can adapt them to fit just about any program.

We worked harder to make them easier to use.

Besides being designed for compatibility with other computers, IBM's new ASCII terminals are more compatible with people.

New 14″ flat screen displays provide a non-glare viewing surface and smooth scrolling. Our 310 and 410 models also offer a choice of 80 or 132 column displays, with crisp character resolution, in green or amber-gold.

What's more, we built the logic into the monitors, making all three models more compact and more reliable.

However, the most important feature of our new displays isn't on the screen, but above it: the IBM name. IBM provides not only a choice of a one or three year warranty, but a tradition of quality, service and support.

It's no wonder these three letters have come to symbolize so much to so many people. And at these prices, they'll be even more in demand. For additional information, contact your IBM Marketing Representative, or call 1-800-IBM-2468 for a supplier near you.

Model 310 Model 410 Model 110

* Prices subject to change without notice.

the state of processing (what constraints had been satisfied) to determine the next steps. This required having rules attempting to match the pattern of goals in progress (on a sponsor's agenda). GoldWorks does not permit matching against agenda items, rather the product matches against assertions. A call for assistance to Gold Hill's support staff, however, revealed how to build frame instances to monitor the goals being matched against, and then have rules that would match against assertions representing slots of these frames. Not only did this increase inferencing capabilities, but it enhanced explanatory capabilities.

GoldWorks' open architecture provides facilities needed to make additions seamlessly to the system. The developers built a program to produce output diagrams for showing configuration, class, and property constraints. The diagrams are easier for some users to understand than equivalent output obtained by browsing through the frame lattice.

Sample results. GoldWorks proved an excellent tool for developing the Expert Configurer. The frame base allowed structuring the hierarchy of configurations describing assemblies, subassemblies, and component parts. Rule sets allowed definition of processing strategies to match configurations to the user's requirements and then to available inventory. Demons and object programming insured only valid combinations. The agenda and sponsor mechanisms made the system responsive to user input and to direct realtime processing of user requests. Finally, the menu and developer's interfaces allowed painless building of the system and interfaces required to communicate with the user.

To successfully penetrate this highly competitive market, expert systems must be as user-friendly as today's best-built PC products. Gold Hill has succeeded in making the GoldWorks menu interface easy to use. For problems that require the user to capture a taxonomic hierarchy or be aware of frame structures, the GoldWorks menu interface is just what is needed. In the sample application, once the structure was defined (for example, configurations consist of classes, which consist of subconfigurations), the developer's interface was useful in building tools to help insulate the user from the underlying frame structure.

Building robust, easy-to-use tools is easy to do in LISP. Its prototype-and-evolve method allows the developer to interact with the potential user to get immediate feedback on ease of use of the interface under development.

ALMOST GOLD

For a PC-based expert system shell, GoldWorks comes close to mainframe capabilities. It has most of the features of high-end tools available on mainframes, such as Inference Corporation's Automated Reasoning Tool (ART) and IntelliCorp's Knowledge Engineering Environment (KEE), but to fully emulate the mainframe shell, it would have to add a few mainframe-like features, including hypothetical reasoning and knowledge-based simulation. Hypothetical reasoning is the ability to have multiple virtual databases so that individual what-ifs can be posed. This is most useful when reasoning requires several possible interpretations. Knowledge-based simulation combines the symbolic reasoning and object programming of expert systems with functions for event creation and scheduling, and collecting simulation results.

The developers built a program to produce output diagrams for showing configuration, class, and property constraints.

GoldWorks is best suited to expert system projects that emphasize knowledge acquisition and inferencing flexibility. It presents a rich menu interface for knowledge engineers who are not LISP programmers. Furthermore, it includes many sophisticated features, such as certainty calculations, explanations of system actions, and multiple inheritance. The novice user can begin with a simplified problem and incrementally build the rule base. The sponsor/agenda structure can be used in a default mode (simple forward and backward chaining); then rules can enable and disable sponsors to give the developer more control over inferencing. GoldWorks provides resource-limited inferencing through its sponsor and agenda structure, using quanta to limit processing done at any level of the sponsor hierarchy.

GoldWorks is further distinguished by its ability to factor knowledge in rule sets. It allows the developer to build large GoldWorks applications where rules are divided into sets that are applicable at different times during processing. This hastens the matching of assertions to a huge knowledge base. The product can easily be enhanced with other sophisticated features; in dependency information associated with assertions, for example, GoldWorks has the foundations for adding truth-maintenance functions. The developer would have to add contradiction-finding and consistency-maintenance functions to make it a full-blown truth-maintenance system.

Gold Hill might consider adding temporal and causal reasoning to widen the scope of GoldWorks' problem-solving. Temporal reasoning is required for planning, scheduling, and problems that require a notion of continuous change, such as expert systems designed for the manufacturing shop floor and other time-critical systems. Causal reasoning, including qualitative model-based reasoning, is required to build systems where limited knowledge in If . . . Then rules is insufficient (where a causal model of the domain is required to specify relations among actions, situations, and events). For July 1988, Gold Hill plans an updated version, featuring full Common LISP, an enhanced screen toolkit, and a graphics interface including active images.

A wish list in these areas might include a screen toolkit that supports full-color, overlapping, sticky and pop-up windows, and a utility that allows the user to build menus and windows interactively, with required commands built automatically by the utility. Graphics, perhaps similar to Microsoft Windows, would be useful in many applications, such as process control. Connections to other PCs, minis, and mainframes would aid database access, including SQL databases.

The layered architecture and open-system approach of GoldWorks makes it easy for both first-time and experienced expert system developers to use. With this ease and its many advanced features, GoldWorks on the PC rivals similar mainframe tools.

Gold Hill Computers
163 Harvard Street
Cambridge, MA 02139
617/492-2071
GoldWorks 1.1: $7,500

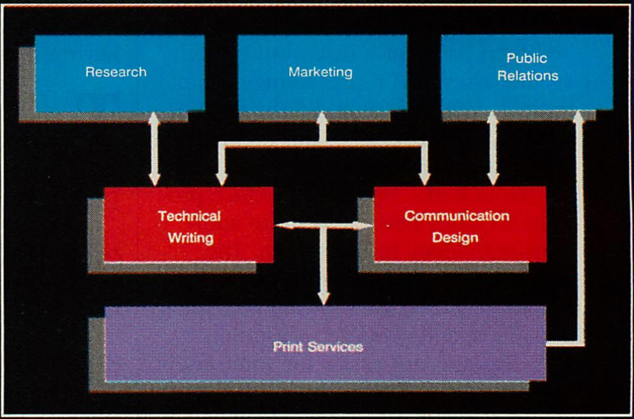
CIRCLE 331 ON READER SERVICE CARD

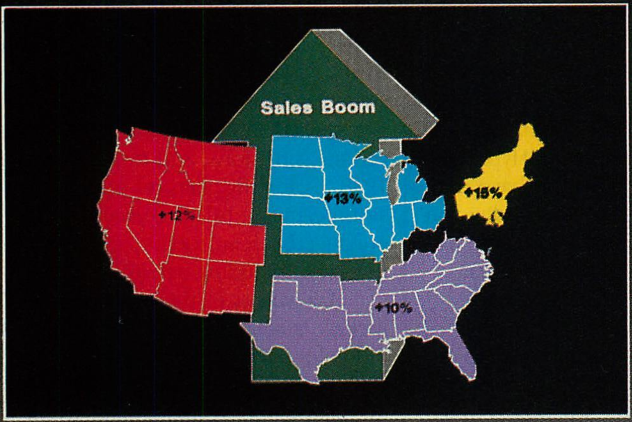
Ken Levine is president of Lekton Inc., a software development firm focusing on commercial applications of artificial intelligence.

The SAS®

Powerful.

The SAS System brings today's power-hungry PC users efficient data management, an easy report generator, customized presentation graphics, superior statistics, and more. You get the strength and flexibility that make SAS software so indispensable on mainframes and minicomputers.

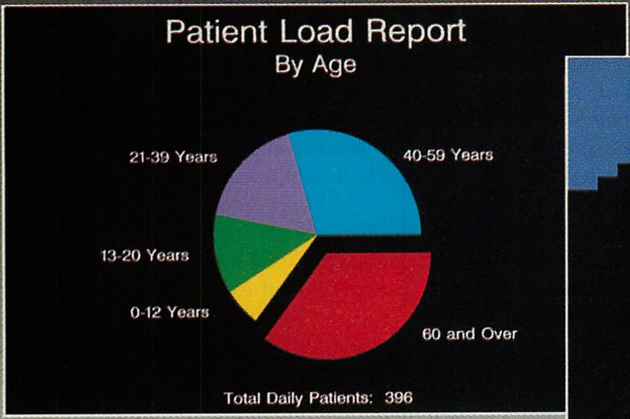


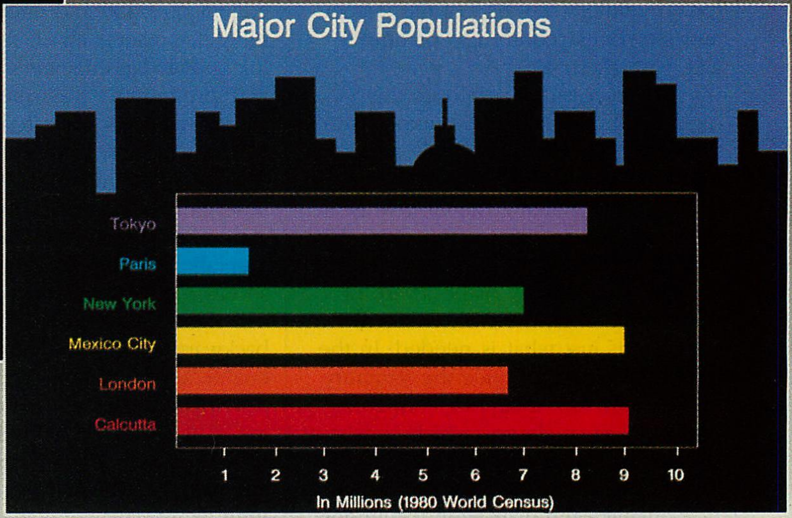


Productive.

The SAS System has integrated applications to use "as is" or customize to fit your needs. Plus a built-in micro-to-host link just for your PC.

Read data in any format from any file including dBASE® and Lotus® 1-2-3®. Then analyze and display your data through interactive windows.





The SAS System runs on the IBM PC AT, XT, and PS/2; IBM 370/30xx/43xx and compatible mainframes; Digital Equipment Corporation's VAX™ series minicomputers and workstations; Data General Corporation's ECLIPSE® MV series; and Prime Computer, Inc.'s 50 series. Not all products are available for all operating systems.

OS/2's Answer to TSRs

Well-defined rules for device monitors and screen pop-ups in OS/2 replace the messy, undocumented shenanigans perpetrated by many DOS TSR utilities.

DAN ROLLINS

In the past few years, pop-up utilities have evolved from a curiosity into a necessity. Today, the power user's PC is fully loaded with a calculator, notepad, outliner, and keyboard macro, all of which can be terminate-and-stay-resident (TSR) utilities. A vital part of any such pop-up utility is a routine that waits for and reacts to the event that triggers the pop-up. This is a complex problem that was never satisfactorily solved under DOS.

OS/2, however, provides a class of programs known as *device monitors* that simplify, standardize, and enhance this process. A device monitor is a program that tracks (or *monitors*) the data stream being processed by a character device driver.

The problem with pop-ups under DOS is that multiple resident programs represent a form of multitasking, a capability that DOS was never meant to support. Even when dormant, a pop-up must perform some processing to examine the events occurring in other processes (for example, keyboard input or the state of the timer) and to decide whether or not to spring into action.

Because DOS does not provide, much less enforce, rules for the interaction of multiple programs, most pop-ups use undocumented DOS functions (interrupt 28H), hardware idiosyncracies (video controller registers), and even hard-coded addresses within DOS (the critical error flag) to perform their magic. No two manufacturers use these in quite the same way, and the resulting difficulties have been described many times: some pop-ups are incompatible with some application programs or have unpleasant side effects (such as crashing the system) when they are used in combination with other pop-ups.

As a multitasking operating system, OS/2 can regulate and oversee multiple concurrent programs. For the first time, developers have a legitimate interface for writing pop-up utilities without underhanded shenanigans. Further, with a few lines of C code in OS/2, a programmer can do a task that takes pages of assembly language, spaghetti logic in DOS.

The OS/2 pop-up interface involves two functions. First, the pop-up must be able to recognize a trigger, perhaps a timer-based event or the pressing of a mouse button. Most pop-ups are triggered when a selected key is pressed. This ability is provided by the keyboard device driver in the form of device-monitor support.

Second, most pop-up utilities need access to the screen when they are triggered. This ability is provided by the OS/2 video system, which allows for

PHOTOGRAPHY • LIGHTSCAPES

FIGURE 1: ***Monitor and Pop-up API Functions***

```
DosMonOpen     obtains a monitor handle to be used in subsequent
               monitor calls.

DosMonOpen( DeviceName, MonHandle );
  char far *DeviceName;        ASCIIZ string, eg, "KBD$"
  unsigned *MonHandle;         Receives the handle


DosMonReg      Setup monitoring buffers and select the logical
               device index.

DosMonReg( MonHandle, InBuf, OutBuf, PositionCode, Index );
  unsigned  MonHandle;         Handle obtained from DosMonOpen()
  char far *Inbuf;             Address of buffer for DosMonRead()
  char far *OutBuf;            Address of buffer for DosMonWrite()
  unsigned  PositionCode;      1=front, 2=back, 0=don't care
  unsigned  Index;             Device-dependent (screen group for KBD$)


DosMonRead     Read (optionally wait for) a data packet from the
               device driver.

DosMonRead( InBuf, WaitFlag, DataBuf, ByteCount );
  char far *InBuf;             Same address as InBuf used in DosMonReg()
  unsigned  WaitFlag;          0=await next packet, 1=no wait
  char far *DataBuf;           Buffer receives incoming data packet
  unsigned *ByteCount;         Entry: size of DataBuf; Return: size of packet
```

```
DosMonWrite    Read (optionally wait for) a data packet from the
               device driver.

DosMonWrite( OutBuf, DataBuf, ByteCount );
  char far *OutBuf;            Same address as OutBuf used in DosMonReg()
  char far *DataBuf;           Buffer containing outgoing data packet
  unsigned  ByteCount;         Size of the packet in DataBuf


DosMonClose    Terminate a monitor.

DosMonClose( MonHandle );
  unsigned  MonHandle;         Handle obtained from DosMonOpen()

VioPopup       Move calling process into foreground and allocate
               a temporary screen.

VioPopup( WaitFlag, 0 );
  unsigned  *WaitFlag;         0=no wait, 1=wait for screen availability
  unsigned  (reserved);        Must be 0

VioEndPopup    Release control of the popup session.  The process
               interrupted by VioPopup() is returned to the
               foreground and its screen is restored.

VioEndPopup( 0 );
  unsigned  (reserved);        Must be 0
```

A program that uses monitors and pop-ups can be written in assembly language or any high-level language that supports OS/2 API calls. The C language protocol for calling the monitor and pop-up API functions is shown here. Prototypes for these and all other OS/2 functions are available in header files that are supplied with Microsoft's OS/2 Software Development Kit.

any program to take control of the screen and keyboard and move itself into the foreground.

In one respect, OS/2 gives pop-up features to every program written for this operating system. The user can press a hot key to toggle between *screen groups* to get to any program (for a definition of screen groups, see "Enter OS/2," Ted Mirecki, November 1987, p. 52). Unless the programs are specifically written to cooperate, however, they will be totally isolated from one another.

Of course, a primary use of a pop-up is to affect the interrupted program in some way. A keyboard macro utility would be useless if it could not feed keystrokes into an application. A pop-up calculator is convenient, but one that can type a result into the document being edited would be more valuable. A realtime spelling checker must be able to see the words you type *as* you type them.

Example programs (see listings) illustrate a keyboard device monitor, video system pop-up control, and a combination of the two.

DEVICE MONITORS

A device monitor is simply a program that uses a set of OS/2 functions to gain access to an I/O stream within a device driver (see figure 1). This program may be written in assembly language or any high-level language that supports OS/2 application program interface (API) calls. The monitor effectively becomes part of the device driver. It examines every piece of information, or *packet*, processed by the driver. The monitor can remove a packet, pass it in its original or modified form, or add new packets to the data stream.

Each device driver can have multiple monitors, which are arranged in a chain. The first monitor receives data from the driver and passes on its output to the next monitor; the last one in the chain passes any output back to the device driver.

The device driver can allow or disallow monitoring. Of the OS/2 device drivers, only the printer (LPT*n*), mouse (MOUSE$), and keyboard (KBD$) provide monitor support. No support is provided for monitoring asynchronous communications (COM*n*) or the real-mode console device (CON). A program can issue an I/O control (IOCTL) function (category 0BH, function 60H) to determine whether a given device supports monitors. For a description of how a device driver implements monitor support, see "Designing Drivers for OS/2, Part 2," by David A. Schmitt (February 1988, p. 136).

A monitor can be a detached process, tracking I/O for other processes, or it can be local to a particular application. Any process can split off a monitoring *thread* to watch for and preprocess keyboard events without affecting other programs. Thus, a monitor can be used as an equivalent of the BASIC ON KEY . . . command, but with more flexibility. For example, a word processor could use a built-in keyboard monitor to handle all command and function keystrokes, removing them from the normal keyboard input stream.

A device monitor is installed and activated in three phases. First, the monitor notifies the driver that it wants to be installed into the monitor chain. Next, it registers its input and output buffers for receiving data from the driver and passing them back. Finally, it monitors the I/O stream by reading from and writing to these buffers.

The monitor is opened with a call to DosMonOpen. This call requires the address of an ASCIIZ device name and returns a monitor handle, which is used during subsequent registration and closing operations. A nonzero return code indicates an error: either an invalid device name was used or the device does not support monitors.

A C program using any OS/2 API calls should include the header files DOSCALLS.H and SUBCALLS.H (see "The Flexible Interface," David A. Schmitt, November 1987, p. 110). The function prototypes in these headers not only check for the correct number

and type of arguments, but also cast parameters into the correct data types. Near pointers are coerced into far pointers—an essential programming shortcut and safety device.

The next step is to register the monitor. The monitor process uses the DosMonReg call to notify the device driver that a monitor is about to become active, to indicate the location and size of two monitoring buffers, and to provide additional information used in setting up the monitoring process.

The monitoring buffers must be large enough to hold the packets passed by the device driver; this size is established when the device driver first creates an empty monitor chain. The monitor process uses an unusual technique to determine this size. When DosMonOpen is called, the first 16-bit word in each buffer must contain the size (in bytes) of the buffer. Should the specified size be too small to contain a packet, the call will return with an error code, and the *second* word of each buffer will contain the size of an I/O packet. Adding 20 bytes to this value gives the actual minimum size of the monitor buffer. The extra 20 bytes are for operating system overhead in buffer management.

After these buffers are registered, the process should not access them again; OS/2 controls all transfer of data to and from these areas. The monitor will receive copies of each packet and will pass the buffer addresses in subsequent API calls, but it must not access the data in the buffers directly.

The fourth parameter to the DosMonReg call is a position code that indicates where this monitor wants to be placed in the chain. The monitor can ask to be first or last, or it may indicate "don't care" to accept any placement. A monitor at the front of the chain has the first chance to modify or remove a packet. It has the most direct control over the data as they come from the device. A monitor at the back of the chain can see (and can modify or remove) any packets inserted by other monitors. It has the most control over the actions of other monitors and over the final appearance of the packet passed back to the driver.

Monitors have one problem reminiscent of a common difficulty of keyboard interrupt handlers used in DOS: two or more processes fighting for a particular keystroke. There is no way to know if a particular hot key has been used by another monitor. The monitor receives no notification of its position in the chain, and it cannot determine whether the source and destination of the packets it handles are those of the driver itself or those of another monitor. The first monitor to ask to be put at the front of the chain actually is placed there. Monitors registered later are always placed later in the chain, in the order they make the request. The ones requesting the end of the chain are treated similarly.

For example, if a keyboard macro program uses Alt-Z as a hot key, it normally removes that key from the data stream. Monitors installed later in the chain never see Alt-Z. This problem has no obvious solution—it is impossible to "steal back the vector."

Several ways to avoid hot-key collisions are available; most of them are the same as those in DOS. The simplest way is to instruct the user that this monitor must be installed before any other monitors. This method obviously breaks down if more than one monitor needs to be first. Another method is to have the user pick a nonconflicting keystroke and configure the program accordingly. A third method is to reconfigure dynamically by initially looking for several hot keys, then nullifying all but the first one to make it into the monitor: the user is told to try all the hot keys and to stick with the first one that works. To increase the chances of selecting a unique keystroke, OS/2 makes it easy to recognize unusual Shift-key combinations or even to use time-stamp sensitivity; for example, the program could look for a double click of a Shift key.

Monitors have a problem reminiscent of keyboard interrupt handlers used in DOS: two or more processes fighting for a keystroke.

The fifth DosMonReg parameter, the index, is used in conjunction with logical devices. Because a character device driver may be handling several logical devices, it needs to know which instance of the driver should be monitored. The meaning of this index can vary from driver to driver. For printer monitors, the index should be 0. For keyboard and mouse monitors, the index should specify which screen group the process wants to monitor.

The process of specifying a screen group involves a separate subplot. When a monitor is executed as a detached process, it is not associated with any screen group. It must learn the ID of the active screen group (for example, the CMD.EXE session from which the DETACH command was executed), and it must use that value as the index. The DosGetInfoSeg call can be used to determine the current screen group.

A program can always pass an arbitrary value in the index parameter. For example, the screen group ID for the session manager menu is 1. The real-mode screen group is 2 (although, as described below, there are problems in monitoring the 3xBox keyboard). The VioPopup session for video pop-up utilities has an index of 3. Such arbitrary values are not documented and may change in the future.

A process can monitor more than one instance of a logical device. A keyboard monitor could keep track of several screen groups; for example, it could provide a cut-and-paste capability between sessions or it could maintain a common set of keyboard macros. To do so, a program would just open the monitor and use the same monitor handle to register two or more separate sets of buffers. Each monitor should execute as an independent thread and should be registered for the desired device index (screen group).

Once the monitor has been opened and registered, the final step is to begin the read/process/write loop. The monitor calls DosMonRead to read incoming data packets from its registered input buffer. Depending on the value of the wait-flag parameter, this call can block the monitor until data are in the buffer, or it can return with an error code. When successful, DosMonRead copies the packet from the input buffer into the monitor's local buffer. The monitor can then examine and process the packet; typically this involves testing for the hot key and reacting appropriately.

A call to DosMonWrite sends packets down the monitor chain; this copies data from the local buffer to the registered output buffer. If a program needs to consume a keystroke, it can read it without writing it. If a program needs to insert keystrokes into the stream, it can create data packets and write them before it reads the next keystroke.

The read/process/write operation should be a tight loop. The monitor code effectively becomes part of the device driver. Thus, lengthy delays must be avoided, such as waiting for a

semaphore or handling any sort of disk I/O. If a printer monitor were to wait on a semaphore, all printing performed by that driver would come to a halt.

If such delays are likely, the monitor should start a thread dedicated solely to the read/write/process loop. If a potentially time-consuming operation is indicated, it should be handled by other threads of the process.

THE KEYSTROKE PACKET

Writing a device monitor requires having access to the device-driver specifications for the layout and content of the data packets passed into the monitor chain. The data packet used by the OS/2 keyboard driver is shown in figure 2. The first word contains monitor flags (see figure 3) that are present in some form in all monitor data packets. For keyboard packets, the first byte contains the hardware-level scan code of the key described by the packet; the driver may choose to examine this byte to recognize the hot key. This word should be set to 0 for packets inserted by the monitor.

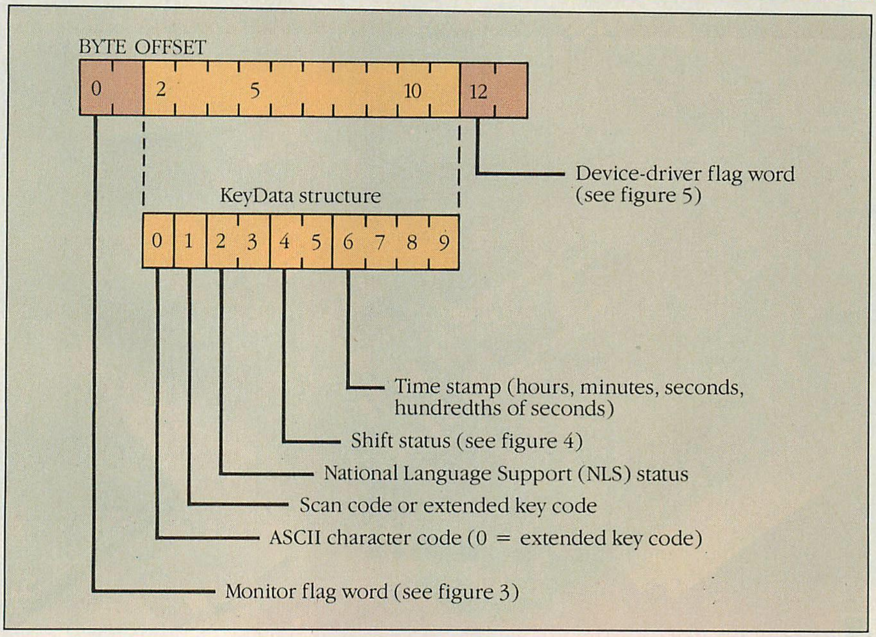
Although the example keyboard monitors presented later in this article ignore it, the flag word can be important in other monitor implementations. Besides information specific to the device driver (this should be documented in the driver specifications), the flag word indicates whether the packet is a normal data packet that is part of the device I/O stream or a request for special processing. Each monitor must pass on each special-request packet after performing whatever processing is appropriate to open or close the device or flush any internal character queues.

The flush request must be handled expeditiously, because the monitor chain does not accept input until the flush packet traverses the entire chain and reaches the driver. The flush operation is not implemented in any of the example drivers because they pass character packets singly without queuing them, so nothing is left to flush.

The body of the packet is similar to the data obtained from a call to the API function KbdCharIn. This 10-byte structure is defined in the SUBCALLS.H include file and is named KeyData. It contains the ASCII character code, scan code, shift status, and information about 2-byte characters from foreign-language code sets.

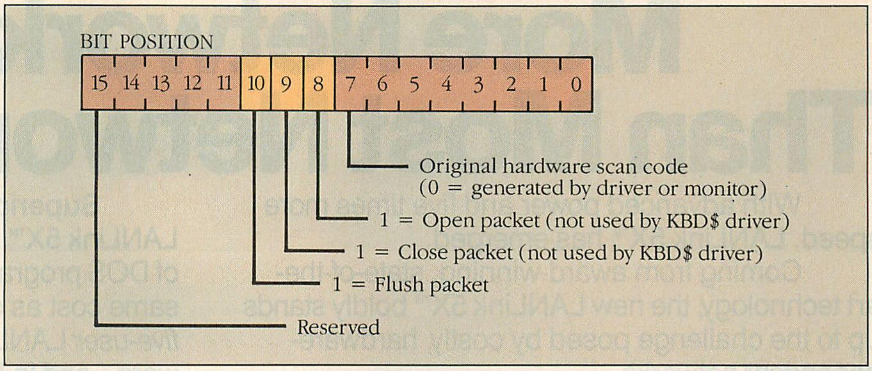
The KeyData.shift_state field (figure 4) is of special interest. A program can monitor for an undefined keystroke by checking for a selected shift state together with some other key.

FIGURE 2: ***Keyboard Data Packet***



The keyboard device driver KBD$ passes the data packet (record) into the monitor chain for each keystroke. A programmer who wants to write a device monitor must know the layout of data packets created by the device driver.

FIGURE 3: ***Monitor Flag Word***



The first word of any driver's data packet contains system-defined and driver-specific flags. A packet with bits 7, 8, or 9 set identifies a request for special processing; such a packet must be passed down the chain by every monitor.

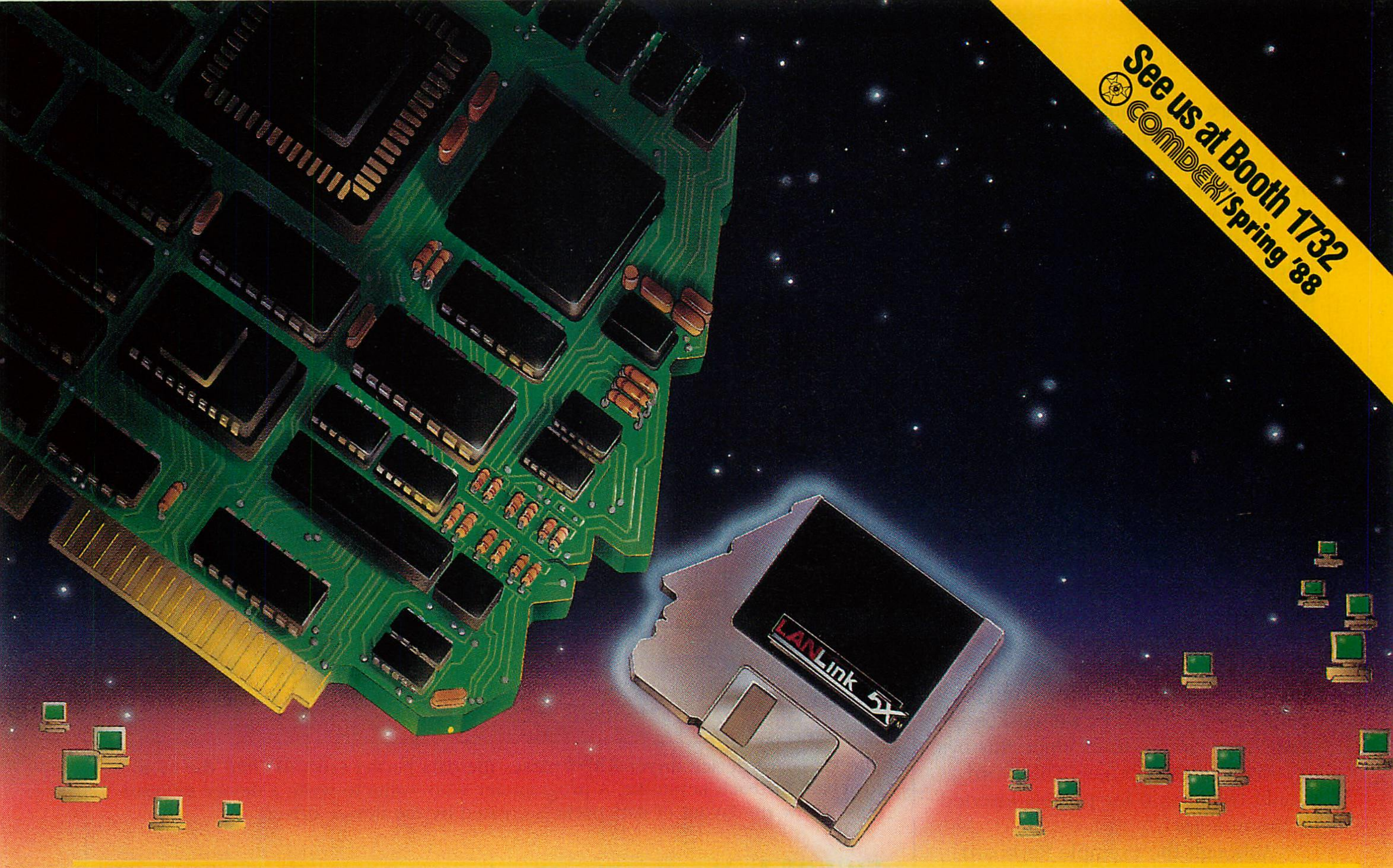
The KeyData.time field shows the time when the packet was generated to the nearest hundredth of a second. Most monitors can ignore this field, but it does present some practical possibilities. For example, a process that monitors both the keyboard and the mouse can use the time field to check for simultaneous events.

The final word of the data packet, KbdDDFlagWord (figure 5), contains significant information. Bits 0 through 5 indicate one of the special types of packets, as shown in table 1. If the pattern in these bits is 3FH, the packet describes an undefined keystroke that a monitor might intercept for its own purposes; a pattern of 13H indicates that the user pressed PrtSc. A keyboard monitor can simulate any of these special codes by inserting a packet with the desired values into the keyboard data stream.

Bit 6 of KbdDDFlagWord is set to 1 on packets generated by the release of a key. By checking this bit, you can differentiate the key break from the key make. In other respects, the make and break packets are identical.

A SIMPLE MONITOR EXAMPLE

CLICKMON.C (listing 1) illustrates the fundamental operations of a keyboard device monitor. It opens the monitor, sets an arbitrarily large buffer size, determines the current screen group, and



More Network Guts Than Most Network Boards

With advanced power and five times more speed, LANLink 5X™ has emerged.

Coming from award-winning, state-of-the-art technology, the new LANLink 5X™ boldly stands up to the challenge posed by costly, hardware-dependent networks.

That takes guts for a software-driven network. But we're confident.

Because we've discovered a high-speed networking channel that's already built into most PCs...the standard parallel port.

The High-Performance Alternative to Network "Boardom." Half a Megabit per second, with over 500 feet between connections. That's the astonishing rate at which LANLink 5X™ transfers data between PCs or PS/2s using ordinary parallel ports.

But you're not restricted to parallel ports. The very same sockets you'd use for modems... connect PCs at 115 Kbps. Plus, you get remote access support and multi-tasking performance under PC-MOS/386™ or PC-DOS.

Superior Results at Inferior Costs. LANLink 5X™ lets network users share data, 1000s of DOS programs, files, and printers. For about the same cost as one network board, you can install a *five-user* LANLink network. Without special hardware...and in less than two hours.

Get Network Guts Without Network Boards. LANLink 5X™ is immediately available and comes with a money-back guarantee. Its price of $595 includes a server and a satellite module plus the network operating system.

Call The Software Link TODAY for complete details on the fastest software-driven network that's ever been available.

CALL: 800/451-LINK

In Georgia:	International/OEM Sales:	Resellers/VARs:	FAX:
404/441-2580	404/263-1006	404/448-5465	404/263-6474

3577 Parkway Lane, Atlanta, GA 30092 Telex 4996147 SWLINK

The Software Link/Canada CALL: 800/387-0453

LANLink 5X™

THE SOFTWARE LINK

DEALER INQUIRIES INVITED

LANLink, LANLink 5X, & PC-MOS/386 are trademarks of The Software Link, Inc. PS/2 & PC-DOS are trademarks of IBM Corp. Prices and technical specifications subject to change.

CIRCLE NO. 196 ON READER SERVICE CARD

registers the buffers. Finally, it drops into the read/process/write loop.

ClickMon is compiled and linked with the command:

cl -Lp -Zp clickmon.c

Depending on the configuration of your system, you may need to specify a -I option to help the compiler find the include files, and you may need to set a **LIB =** variable into the environment to help the linker find the libraries. The **-Zp** compiler option prevents the insertion of slack bytes to align on word and double-word boundaries; this option should be used in nearly all programs that use OS/2 kernel calls, because many system structures are defined with byte-level alignment.

After successful compilation, the monitor is executed with the following command:

detach clickmon

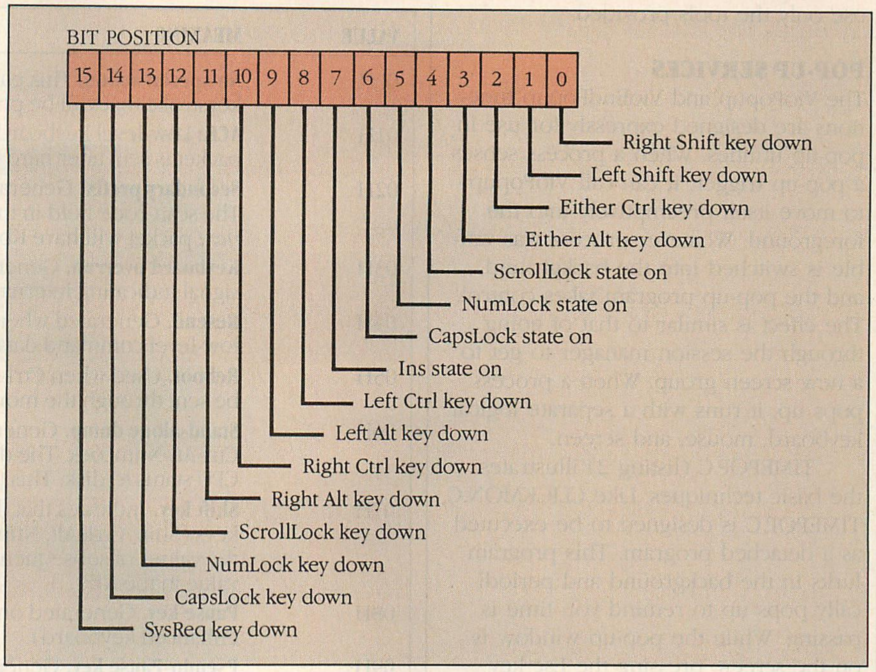
ClickMon simply beeps the speaker each time the keys are pressed. It closes the monitor and terminates whenever the Esc key is pressed. Experimentation with ClickMon shows why such a simple exit mechanism is provided—a detached process must supply its own exit or remain resident permanently. That can cause undesirable side effects as a detached program is developed. For example, you cannot delete or rename the running version, so you cannot create another version with the same name. Users should also appreciate the ability to deinstall monitors cleanly after they have outlived their usefulness; under DOS, in most cases, that was possible only by rebooting the system.

ClickMon pares off as much complexity as possible, making it perfect for experimental purposes. In the DosMonReg call, you might try to hard-code an index value of 1. Now ClickMon has no effect in the screen group where it is started. When you hot-key into the session manager menu (Ctrl-Esc), however, notice that the keyboard is clicking in this screen group.

If screen group 2 (the real-mode session) is monitored, some curious behavior becomes apparent. Each keystroke causes a click, indicating that the monitor is reading correctly, but the write operation seems to fail because the keystroke never makes it down the chain to the 3xBox application. The official word from Microsoft is that device monitors are not supported at all in real mode.

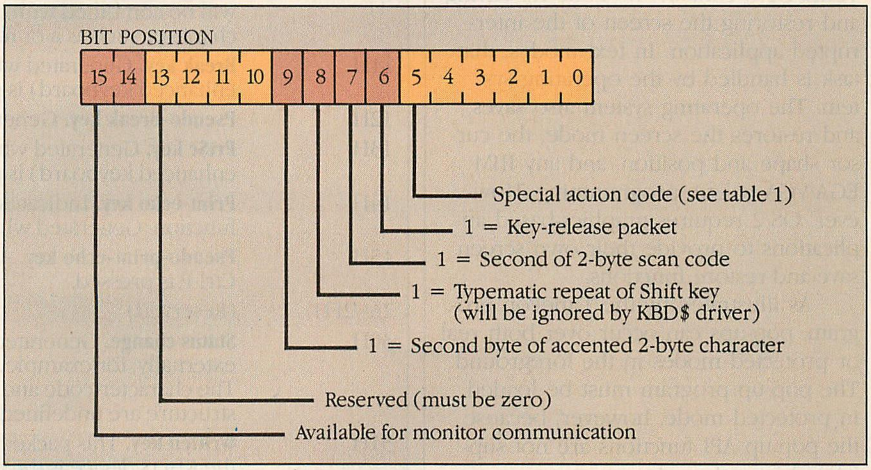
ClickMon clicks on all keystrokes, including the Shift key. It is pro-

FIGURE 4: ***Shift Status Word***



The information in the shift status word is the same as that in bytes 40:17 and 40:18 of the BIOS data area. This information is sent in every packet so that it reflects the status when the key was pressed, not when the monitor processes it.

FIGURE 5: ***Keyboard Device Driver Flag Word***



The flag word marks packets that need special processing, such as combining several packets into one character (which is used in national language support).

grammed to ignore the release of a key, but it also could monitor for that. Furthermore, ClickMon responds to undefined keystrokes, such as Alt-NumLock. In other words, you can monitor nearly every possible low-level keyboard event. The exceptions include the session-manager hot keys (Ctrl-Esc and Alt-Esc) and Ctrl-Alt-Del; these are handled by the device driver or passed directly to the session manager and never make it into the monitor chain.

Using just the tools of a simple keyboard monitor, along with the standard C library functions, you can extend the monitor to carry out such tasks as logging keystrokes to a disk file, remapping areas of the keyboard, performing realtime spelling checking, and intercepting keystrokes that might cause problems (such as Ctrl-Break or Ctrl-PrtSc).

ClickMon lacks the ability to act like the archetypal pop-up; programmers need a way to construct fancy windows and interact with the user. In real mode, a program can just start writing to the screen; in protected

mode, it first must politely ask the operating system's permission and then use only the tools provided.

POP-UP SERVICES

The VioPopup and VioEndPopup functions are designed expressly for use in pop-up utilities. When a process senses a pop-up trigger, it can call VioPopup to move itself preemptively into the foreground. Whatever process was visible is switched into the background, and the pop-up program takes control. The effect is similar to that of going through the session manager to get to a new screen group. When a process pops up, it runs with a separate logical keyboard, mouse, and screen.

TIMEPOP.C (listing 2) illustrates the basic techniques. Like CLICKMON.C, TIMEPOP.C is designed to be executed as a detached program. This program lurks in the background and periodically pops up to remind you time is passing. While the pop-up window is on the screen, pressing the Esc key deinstalls the program; any other key closes the pop-up window, restarts the timing interval, and returns to the interrupted task.

Notice how short the program is. TIMEPOP.C contains no code for saving and restoring the screen of the interrupted application. In text modes, that task is handled by the operating system. The operating system also saves and restores the screen mode, the cursor shape and position, and any IBM EGA/VGA palette programming. However, OS/2 requires graphics-based applications to provide their own screen save-and-restore functions.

As illustrated by the TimePop program, pop-ups can occur over both real or protected modes in the foreground. The pop-up program must be loaded in protected mode, however, because the pop-up API functions are not supported in real mode.

A program is limited in what it can do while popped up. For instance, it cannot access either the logical or physical video buffers directly but must use the API Vio calls. (For a comparison of all three methods of video display, see "The Flexible Interface," David A. Schmitt, November 1987, p. 110.) Fortunately, the performance of the OS/2 video calls is quite good—a far cry from the sluggish real-mode BIOS and DOS screen handling.

In general, a program can write text and attributes, scroll or clear the screen, set the cursor position, and get current screen-mode information. It cannot change the screen mode, reprogram the EGA/VGA palette, install a video subsystem, redefine a font, print the screen, or perform other more advanced tasks.

One important limitation is that only one VioPopup can be active at a time. There is exactly one VioPopup session. If another process controls the VioPopup screen, any other program is barred from access to that screen.

In the prerelease version of OS/2, VioPopup's operation was seriously flawed. This version always selected an 80-by-25 text mode, cleared the screen, and homed the cursor. Of course, that eliminated a lot of what pop-ups are all

TABLE 1: ***Special Action Codes in KbdDDFlagWord***

VALUE	MEANING
00H	**No special action.** This packet contains a normal or extended-ASCII keystroke that will be placed into the keyboard input buffer (KIB).
01H	**ACK:** Low-level keyboard acknowledgement; for example, this packet is sent after hard reset.
02H	**Secondary prefix.** Generated by enhanced keyboard as a prefix. The scan-code field in the KeyData structure is usually 0EH. The *next* packet will have KbdDDFlagWord bit 7 set.
03H	**Keyboard overrun.** Generated when the keyboard hardware issues a signal indicating internal buffer overflow.
04H	**Resend.** Generated when the keyboard hardware requests that low-level command data be resent.
05H	**Reboot.** Used when Ctrl-Alt-Del is pressed. This packet will *not* be sent through the monitor chain.
06H	**Stand-alone dump.** Generated on the second consecutive press of Ctrl-Atl-NumLock. The driver is supposed to dump memory and CPU status to disk. This packet is not sent to the monitor chain.
07H	**Shift key.** Indicates that the packet is one of the defined shift keys (Shift, Ctrl, Alt, NumLock, etc.). The packet will affect the values of subsequent shift status fields, but it will not put a value in the KIB.
08H	**Pause key.** Generated on Ctrl-NumLock (or Pause on the enhanced keyboard).
09H	**Pseudo-Pause key.** Generated when Ctrl-S is pressed.
0AH	**Wake-up key.** Indicates completion of previous Pause action. The keystroke is not placed into the KIB.
0B–0F	(Reserved)
10H	**Accent key.** This packet contains an accent key, as defined in the current keyboard translation table. Setting KbdDDFlagWord bit 9 will generate the accent character itself. Otherwise, this packet will be combined with the *next* packet to generate an accented character such as **à** or **ñ**.
11H	**Break key.** Generated when Ctrl-ScrollLock (or Ctrl-Break on the enhanced keyboard) is pressed.
12H	**Pseudo-Break key.** Generated when Ctrl-C is pressed.
13H	**PrtSc key.** Generated whenever Shift-PrtSc (or PrintScreen on the enhanced keyboard) is pressed.
14H	**Print-echo key.** Indicates toggle of screen-to-printer echo function. Generated when Ctrl-PrtSc is pressed.
15H	**Pseudo-print-echo key.** As above, but it is generated whenever Ctrl-P is pressed.
16–2FH	(Reserved)
30H	**Status change.** Generated when keyboard shift status is modified externally; for example, via KbdSetStatus or DosDevIOCtl. The character-code and scan-code fields in the KeyData structure are undefined and should be ignored.
31H	**Written key.** This packet is generated by a DosWrite call when the KBD$ device is the destination. The KeyData packet will always be placed into the KIB.
32–3EH	(Reserved)
3FH	**Undefined.** Generated by a keystroke that cannot be translated by the driver. This value is used on packets containing undefined keystrokes such as Alt-PrtSc, Ctrl-Shift-Z, and Alt-Del.

The values in bits 0 through 5 of the device driver flag word (see figure 5) identify the kind of special processing needed by certain keyboard packets.

about. The production versions from both Microsoft and IBM provide a "transparent" option that allows you to overlay a smaller window over the background text. It also provides access to the data on the overlaid screen (consider how many pop-up utilities gain information from the cursor position and the text beneath the cursor).

PUTTING IT ALL TOGETHER

The CHARMENU.C program (listing 3) juxtaposes the concepts of device monitors and video system pop-up control. Besides illustrating the tools in a real-life program, it is a useful utility.

CharMenu monitors the keyboard, looking for either of two hot keys—Alt-C or Alt-Spacebar. When Alt-C is recognized, it pops up a menu of hard-to-type ASCII characters, such as box-drawing characters, smiling faces, and Greek characters. The cursor keys point to the desired character, and pressing Enter selects it. This causes the character to be inserted into the keyboard data stream, as if a series of Alt-NumPad keystrokes had been given.

Alt-Spacebar is a shortcut. It inserts the most recently selected character into the data stream without displaying the menu. This feature is most handy after a horizontal bar character (ASCII 196 or 205) has been selected. Just press Alt-Spacebar and let it repeat across the screen.

Like ClickMon, CharMenu begins by opening the monitor for the KBD$ device. The monitor handle is saved and used in subsequent monitor calls. The next step is a more elaborate, generalized version of the registration process. First, the program checks for the correct buffer size by making a call to DosMonReg, specifying a buffer size of 0. When this call returns with an error, the second word in the buffer contains the required data packet size. The next call, requesting 20 bytes more than the packet size, then succeeds in registering the buffers.

The program then goes into the monitoring loop. The DosMonRead call is performed with the wait option, so execution is blocked until a packet comes in. Once the packet has been read, it is checked to see if it contains either of the hot keys.

To sense Alt-C, the program compares the packet's character and scan code with the desired values. However, the second hot key, Alt-Spacebar, is an undefined key—Spacebar always returns the same character and scan code, regardless of the state of the Shift keys. CharMenu tests for the keystroke by checking for the character code and then testing for the Alt-Shift key bit in the packet's Shift-key flags.

When an Alt-Spacebar packet arrives, the program writes a packet containing the most recently selected character. Several fields are set to 0 on such an inserted packet, much as is done on packets that are generated when Alt is pressed followed by a series of NumPad keys.

CharMenu looks for both the press and the release packets of its hot keys, removing both from the I/O stream. For a key-press packet, the program takes action and discards the packet without passing it down the chain. When a release packet is encountered, the program ignores and discards it. This prevents spurious key-release packets from being sent down the chain. Although such extra packets pose no problems to the driver, they could confuse intervening monitors.

When CharMenu sees an Alt-C, it calls the function DoMenu. This function illustrates a more complex use of VioPopup. It attempts a no-wait pop-up.

Through Rain, Snow, Sleet, Hail, IBM, Novell, 3Com and Banyan LANs, Remote Sites, Bridges and Public E-Mail Systems... The Network Courier *delivers!*

If You're A LAN User, Chances Are You've Already Heard Of The Network Courier. After All, The Network Courier Has Become Accepted As The Industry-Standard LAN Electronic Mail Package. But, You May Not Yet Have Heard Of Its Power To Take Your Messages Outside Of An Individual LAN.

We've added the links that can make LANs a solid part of your corporate-wide E-Mail network. The new Inter-Network Courier provides transparent connectivity between all types of NetBIOS compatible LANs, including IBM, 3COM, Novell, Banyan and Ungermann-Bass. There is even a Network Courier module that links LAN users and IBM PROFS users together.

To go beyond your internal organization, Network Courier offers gateways to the world of public E-Mail, through systems such as MCI Mail.

Our recently completed Remote User version now allows even the most remote individual PC to call into your LAN "post office," via modem, from anywhere.

And, whichever version you're using...whether it's user-to-user E-Mail within a LAN, Inter-Network, or via public E-Mail system...your users can attach anything from spreadsheets to DOS programs along with their messages.

For More Information On The Network Courier, Phone: 800-663-8935
Dealer Inquiries Invited.

Of Course, Connectivity Is Only Part Of The Story. Here Are Some Of The Features That Have Made The Network Courier User Base Look Like A Who's Who Of Fortune 500 Companies:

- "pop-up" mail monitor so you don't have to leave your current application to send and receive mail
- Easy to use intuitive interface. On-line help provided.
- Attach up to ten DOS files to a message. Cut and Paste from any ASCII file.
- Multiple priority levels, including registered (receipt delivery) mail
- Compatible with CCITT X.400 international E-Mail standard
- Mail transfers transparently between LANs of different types (e.g., Novell and 3Com)
- Gateways to mainframe mail systems
- Gateways to public E-Mail systems
- Supports dial-in from remote users in the field

More experts have endorsed The Network Courier than any other LAN product. The Network Courier is licensed and/or re-sold by AT&T, Tandy, Ungermann-Bass, ACT, Univation and many others.

Consumers Software is the original developer of The Spreadsheet Auditor, and the producer of Duet, the new on-screen printer control software that offers sideways background printing.

THE Network COURIER™

Consumers Software Inc.
736 Chestnut Street, Santa Cruz, CA 95060

CIRCLE NO. 168 ON READER SERVICE CARD

If this fails, the program beeps out a tune and resumes without taking control of the screen. This sequence may be needed if some program has locked the screen or if another VioPopup session is already active. This precaution is probably unnecessary here—CharMenu is not active while any VioPopup is active, because it monitors only its original screen group. It is a good practice to use this technique, however, especially in timer-triggered pop-ups that are not tied to any one logical device.

Next, the menu is drawn, and the interactive portion of the program begins. This program uses standard techniques for handling a cursor-bar menu. Perhaps most interesting is the use of VioWtrNAttr to remove and redraw the cursor bar highlight with each keystroke. The beauty of this function is that the text at the cursor does not need to be rewritten; only its video attribute needs to be changed.

The arrow keys move the highlight. The Enter key selects the currently highlighted character, Esc aborts the menu without selecting a character, and Q causes CharMenu to terminate itself. Whatever key is pressed, the VioEndPopup service is called to release the pop-up screen and return the interrupted program to the foreground.

An examination of the interactive code reveals that the standard library function, Getch, is used in preference to the KbdCharIn call. This illustrates an important facet of OS/2 programming: just because a kernel call exists does not mean that it must be used. Getch is convenient, and it works, so there is no reason to use the lower-level system call.

This simplicity comes at some cost in performance, however. Getch eventually calls KbdCharIn, which takes longer than if the program called the API function directly. This could be a consideration in extremely time-critical applications. The greater portability of standard library functions is not an issue, because an OS/2 pop-up utility is specific to its operating environment, regardless of which function it uses.

FLEXIBLE MONITORS

As mentioned earlier, a monitor can be used as an equivalent of the BASIC ON KEY . . . command, but one with far more flexibility. Because a keyboard monitor sees all key presses and releases, it can allow an application to act on a variety of unusual keyboard events—a double-click of a Shift key or a super-shift combination such as LeftShift-RightShift-Z. A keyboard monitor can perform useful services, such as speeding up typematic repeats and eliminating keyboard run-on. An intelligent keyboard monitor can recognize and mask differences between different types of keyboards.

Another idea is to cross-monitor two different devices. A keyboard monitor could be used to simulate mouse events or vice versa. By monitoring both devices, a process can take special actions for unique combinations of keyboard and mouse events. For example, a program could detect mouse movement or a button press that occurs while a Shift key is pressed and differentiate it from normal mouse events.

In OS/2, IBM and Microsoft have designed a system that incorporates the best features of DOS while providing multitasking and multithreaded operation. They moved all the old BIOS operations under the operating system's umbrella, made them faster, and provided new and better tools for creating the next generation of applications. The OS/2 pop-up tools illustrate the extensive planning and creativity that went into the new operating system.

Dan Rollins, a freelance technical writer, is the author of the Flambeaux Software Help! series and the Norton Online Guide to OS/2.

ZIM®

Application Development

Fully Integrated:
* Interactive forms design
* Application generator
* Menu generator
* Active dictionary
* Advanced debugging aids
* Programming language interface
* Report writer / specifier
* Entity-relational (ER) dbms
* Runtime systems
* Multi-level security

Completely Portable:
* PC- MS-DOS
* Novell, MS-Net
* UNIX, QNX, XENIX
* VAX/VMS
* IBM VM/CMS

Advanced Architecture:
* SQL
* A-I based
* Case modelling

ZIM 3.0 is the most comprehensive 4GL application development system available today, offering complete micro-to-mainframe application portability (with no code changes). ZIM's elegant, procedural language permits everything from simple queries through to complex database operations and provides capabilities for comprehensive data extraction, manipulation and presentation.

ZIM has won accolades from professional application developers worldwide who appreciate a barrier-free development environment that delivers optimum performance on every major technological platform.

ZIM is supported by an ever-growing family of companion products like NIVA's Guide to ZIM (a novice-to-expert learning and reference text), ZIM/ISQL which allows queries based on SQL syntax, Runtime and Query-Runtime systems and ZIMPLE, an add-on module that permits easy use, allows end-users to create their own simple applications, ad hoc queries or paint their own reports (used with ZIM or Query Runtime).

Around the world, from Australia to Europe, ZIM Service Centers and Distributors stand ready to supply and assist any ZIM user.

Already an integral part of data management in national governments, international banks, financial corporations and thousands of other applications ZIM welcomes you to the world of serious application development.

For more information call today!

1-800-267-9972 (U.S.)
(613) 727-1397 (Canada and Overseas)

Distributor and VAR Inquiries Welcome

ZANTHE

Designed With Professional Developers In Mind!

LISTING 1: CLICKMON.C

```
/* ClickMon.C   Illustration of a simple OS/2 keyboard monitor.
   Monitors the keyboard and makes a click on each key press.

   To compile:  CL -Lp -Zp clickmon.c
   To execute:  DETACH CLICKMON
   To remove:   Press [Esc] while in its screen group.
*/

#include <doscalls.h>
#include <subcalls.h>

#define ESCAPE    27
#define RELEASE  0x40      /* Kp.ddflags bit indicates key release */

struct KeyPacket {         /* keyboard monitor data record */
   unsigned  monflags;
   char      ascii;
   char      scan;
   char      idontcare[8];
   unsigned  ddflags;
} Kp;                        /* allocate storage for one packet */

char InBuf[128];             /* buffers used for keyboard monitoring */
char OutBuf[128];            /* hard-coding the size for simplicity */

main()
{
   unsigned monhandle, kplen, index;

   DOSMONOPEN ( "KBD$", &monhandle );      /* get a monitor handle */

   InBuf[0]=128;  InBuf[1]=0;          /* setup size for DOSMONREG */
   OutBuf[0]=128; OutBuf[1]=0;
   index = CurGrp();                 /* get current screen group id */

   DOSMONREG( monhandle, InBuf, OutBuf, 1, index );

   /* ============ Monitor Read/Process/Write Loop =========== */
   while ( 1 ) {
      kplen = sizeof(Kp);
      DOSMONREAD( InBuf, 0, (char *)&Kp, &kplen );  /* read packet */
      if ( Kp.ascii == ESCAPE ) {        /* ESC to kill the monitor */
         DOSMONCLOSE( monhandle );
         DOSEXIT ( 1, 0 );
      }
      if ( (Kp.ddflags & RELEASE) == 0 )    /* if not a key release */
         DOSBEEP( 250, 35 );                /* packet, make a click */

      DOSMONWRITE( OutBuf, (char *)&Kp, kplen );   /* write packet */
   }
}

/* =========== This obtains the ID of the current screen group ==
   The byte at offset 24 in the InfoSeg is the current screen group.
   This function just extracts that one value.
*/

CurGrp()
{
   unsigned infoseg, localseg, cg;
   char far *gdt;

   DOSGETINFOSEG( &infoseg, &localseg );
   gdt = (char far *)( (long)infoseg << 16 );
   return( gdt[24] );
}
```

LISTING 2: TIMEPOP.C

```
/* TimePop.C   Illustration of a simple OS/2 VioPopup application.
   Pops up and displays a message once every so often

   To compile:  CL -Lp -Zp timepop.c
   To execute:  DETACH TIMEPOP
   To remove:   Press [Esc] when it is popped up.
*/

#include <doscalls.h>
#include <subcalls.h>

char Msg[] = "Another interval has passed.  Press any key ...";

main()
{
   unsigned waitflag, err;
   long seconds = 20;

   while(1) {
      err=DOSSLEEP( seconds * 1000L );   /* interval in millisecs */

      waitflag=1;                          /* wait if not ready */
      err=VIOPOPUP( &waitflag, 0 );

      VIOSETCURPOS( 12, 15, 0);
      VIOWRTTTY( Msg, strlen( Msg ), 0 );

      if ( getch() == 27 ) DOSEXIT ( 1,0 );   /* Esc kills it */
      VIOENDPOPUP( 0 );
   }
}
```

LISTING 3: CHARMENU.C

```
/* CharMenu.C    OS/2 Popup Menu for Characters
   Pops up and displays a menu of hard-to-type characters.

   Press Alt-C, move the cursor to a character, and press Enter.
   Press Alt-spacebar to reuse the previous selection.

   To compile:  CL -Lp -Zp timepop.c
   To execute:  DETACH CHARMENU
   To remove:   Press [Q] while it is popped up.
*/

#include <doscalls.h>
#include <subcalls.h>
#include <malloc.h>

#define FRONT        1   /* position code for DosMonReg */
#define OVRHD       20   /* overhead needed in monitor buffers */
#define HOT_CHAR_1   0   /* Alt-C char code and scan code */
#define HOT_SCAN_1  46
#define HOT_CHAR_2  32   /* the spacebar */
#define ALT          8   /* bit in Kp.shift, set when Alt is down */
#define RELEASE   0x40   /* Kp.ddflags bit indicates key release */
#define MENU_NO_KEY  0   /* returned from DoMenu() and DoPopup() */
#define MENU_QUIT   -1
#define NORMAL  0x07     /* screen attributes used */
#define REVERSE 0x70
#define BOLD    0x0F

struct KeyPacket {              /* keyboard monitor data record */
   unsigned       monflags;
   unsigned char ascii;         /* a translated ASCII value */
   unsigned char scan;          /* a translated scan code value */
   unsigned       nls;          /* National Language Support stuff */
   unsigned       shift;        /* shift-key flags */
   unsigned long keytime;
   unsigned       ddflags;      /* indicates make/break, et.al. */
} Kp;                           /* allocate one instance as Kp  */

char far *MonInBuf;             /* addresses of monitor buffers */
char far *MonOutBuf;
int  CurX=0, CurY=0;            /* initial menu defaults */
unsigned char CurChar=32;

main()
{
   unsigned monhandle, kplen, grpid, retcode, err;
   unsigned tmpib[2], tmpob[2];          /* used to determine size */

   DOSMONOPEN ( "KBD$", &monhandle );      /* get a monitor handle */
   tmpib[0] = tmpob[0] = 0;
   grpid = CurGrp();        /* get current screen group for 'index' */
   DOSMONREG( monhandle, (char *)tmpib,(char *)tmpob, FRONT, grpid );
/* assume failure and use resulting values as the correct size */

MonInBuf  = (char far *)malloc( tmpib[1] + OVRHD );
MonOutBuf = (char far *)malloc( tmpob[1] + OVRHD );
MonInBuf[0] =tmpib[1]+OVRHD; MonInBuf[1] =0; /* set first WORD */
MonOutBuf[0]=tmpob[1]+OVRHD; MonOutBuf[1]=0; /* to buffer size */
```

```
err = DOSMONREG( monhandle, MonInBuf, MonOutBuf, FRONT, grpid );

InstallMsg( err );   /* announce installation sucess or failure */
if ( err != 0 )
   DOSEXIT( 1, 0 );          /* terminate if unable to register */

/* ------------- Monitor Read/Process/Write Loop ----------- */
while ( 1 ) {
   kplen = sizeof(Kp);                      /* read a key packet */
   DOSMONREAD( MonInBuf, 0, (char *)&Kp, &kplen );
                                      /* ignore all key releases */
   if ( (Kp.ddflags & RELEASE) == 0 ) {
      if ( (Kp.ascii == HOT_CHAR_1) && (Kp.scan == HOT_SCAN_1) ) {
         retcode = DoPopup();
         if (retcode == MENU_NO_KEY )            /* Esc in menu? */
            continue;                         /* yes, no action */
         if (retcode == MENU_QUIT ) {             /* Q in menu? */
            DOSMONCLOSE( monhandle );
            DOSEXIT ( 1, 0 );               /* terminate CharMenu */
         }
                                          /* else, write the key */
         Kp.ascii = retcode; Kp.scan = 0;
         Kp.ddflags = Kp.monflags = 0;
         DOSMONWRITE( MonOutBuf, (char *)&Kp, kplen );
         continue;                               /* read next key */
      }
      /* Alt-spacebar is an undefined packet.  If the character is
         a space and the Alt-key is down, we use it.
      */

      if ( (Kp.ascii==HOT_CHAR_2) && ((Kp.shift & ALT)==ALT) ) {
         Kp.ascii = CurChar;  Kp.scan = 0;
         Kp.ddflags = Kp.monflags = 0;
         DOSMONWRITE( MonOutBuf, (char *)&Kp, kplen );
         continue;                               /* read next key */
      }
   }           /* all other packets get written immediately */
   DOSMONWRITE( MonOutBuf, (char *)&Kp, kplen );
 }
}
```

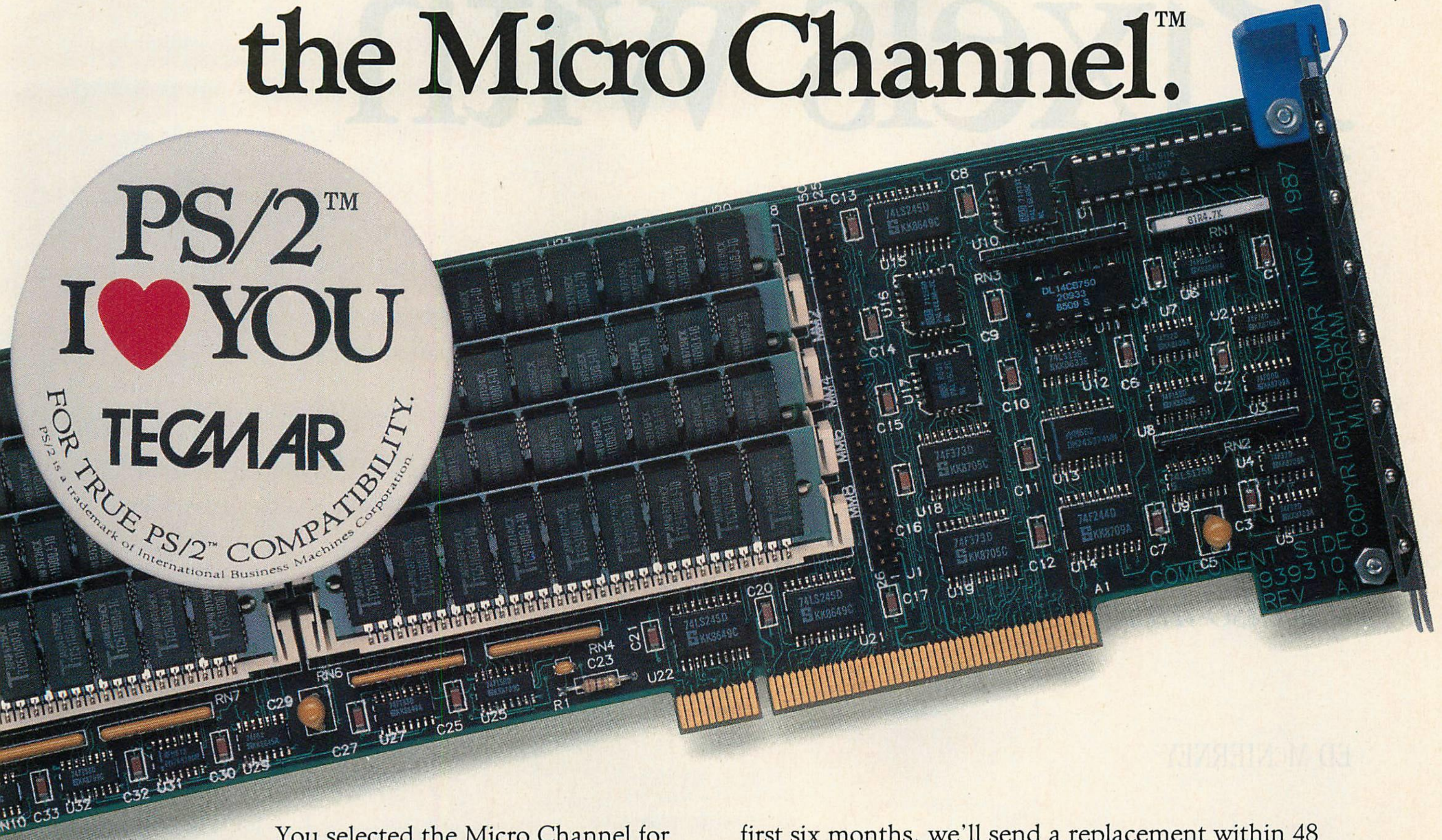
```
/* ========================== Code and data for the popup menu == */
char MTop[]= "┌──────────────────┐";
char MMid[]= "│                  │";
char MBtm[]= "└──────────────────┘";
char MMsg1[]=" Enter to select and exit ";
char MMsg2[]=" Esc    to exit           ";
char MMsg3[]=" Q      to remove CharMenu ";


#define MENU_HI   20
#define MENU_WIDE 8


unsigned char CharArray[ MENU_HI ][ MENU_WIDE ] = {
 { 0xcd, 0xc9, 0xcb, 0xbb, 0xda, 0xc2, 0xbf, 0xc4 }, /* Box-drawing */
 { 0xba, 0xcc, 0xce, 0xb9, 0xc3, 0xc5, 0xb4, 0xb3 }, /* characters  */
 { 0xdb, 0xc8, 0xca, 0xbc, 0xc0, 0xc1, 0xd9, 0xdc }, /* 176 - 223   */
 { 0xb2, 0xd6, 0xd2, 0xb7, 0xd3, 0xd1, 0xb8, 0xdd }, /* in a sane   */
 { 0xb1, 0xc7, 0xd7, 0xb6, 0xc6, 0xd8, 0xb5, 0xde }, /* layout      */
 { 0xb0, 0xd3, 0xd0, 0xbd, 0xd4, 0xcf, 0xbe, 0xdf },
 { 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07 }, /* 0 - 31 */
 { 0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f },
 { 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17 },
 { 0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f },
 { 0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87 }, /* 128-175 */
 { 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f },
 { 0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97 },
 { 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f },
 { 0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7 },
 { 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf },
 { 0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7 }, /* 224-255 */
 { 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef },
 { 0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7 },
 { 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff }
};

/* ====== Popup, draw menu, get key, release popup == */

DoPopup()
{
```

```
    unsigned waitflag, row, col, keycode, err;
    char attr;

    waitflag = 0;                   /* no wait */
    err=VIOPOPUP( &waitflag, 0);    /* get popup control */
    if ( err ) {                    /* can't popup now */
        DOSBEEP( 400,100 );         /* play a "sorry 'bout that" tune */
        DOSBEEP( 300,200 );
        DOSBEEP( 200,300 );
        return( MENU_NO_KEY );
    }
    DrawMenu();                     /* display the menu */
    keycode = DoCharMenu();         /* get interactive input */
    VIOENDPOPUP( 0 );               /* release popup control */
    return( keycode );            /* send selected code back to caller */
}

/* ========================== This draws the menu == */
DrawMenu()
{
    char row, col, attr;
    struct CursorData cd;           /* from subcalls.h */

    attr = NORMAL;                  /* write the box around the menu */
    VIOWRTCHARSTRATT( MTop, strlen(MTop), 0, 0, &attr, 0 );
    for ( row=1; row <= MENU_HI; row++ ) {
        VIOWRTCHARSTRATT( MMid, strlen(MMid), row, 0, &attr, 0 );
    }

    VIOWRTCHARSTRATT( MBtm, strlen(MBtm), MENU_HI+1, 0, &attr, 0 );

    attr = BOLD;
    VIOWRTCHARSTRATT( MMsg1, strlen(MMsg1), MENU_HI+2, 0, &attr, 0 );
    VIOWRTCHARSTRATT( MMsg2, strlen(MMsg2), MENU_HI+3, 0, &attr, 0 );
    VIOWRTCHARSTRATT( MMsg3, strlen(MMsg3), MENU_HI+4, 0, &attr, 0 );
    attr = BOLD;

    for ( row=0; row < MENU_HI; row++ ) {     /* write menu contents */
        for ( col=0; col < 8; col++ ) {
            VIOWRTCHARSTRATT( &CharArray[row][col], 1,
                                        row+1, (col*3)+2, &attr, 0);
        }
    }

    VIOGETCURTYPE( &cd, 0 );
    cd.cur_attribute = -1;           /* turn off the blinking cursor */
    VIOSETCURTYPE( &cd, 0 );
}

/* == This handles menu input; returns selection */

int DoCharMenu()
{
    char attr;
    char tmp[20];
    int ch;

    while ( 1 ) {
        CurChar = CharArray[CurY][CurX];

        /* ------------ display current value in hex and decimal -- */
        sprintf( tmp, " %3u %2Xh ", CurChar, CurChar );
        attr=NORMAL;
        VIOWRTCHARSTRATT( tmp, strlen(tmp), MENU_HI+1, 14 , &attr, 0 );

        /* -------------------- show cursor bar in reverse video -- */
        attr = REVERSE;
        VIOWRTNATTR( &attr, 3, 1+CurY, 1+(CurX*3), 0 );

        /* ------------------------------- wait for a keystroke -- */

        ch=getch();                   /* handy getch() technique:   */
        if (ch == 0) ch = -getch();   /* make X-ASCII keys negative */

        /* ---------------------------------- remove cursor bar -- */
        attr = BOLD;
        VIOWRTNATTR( &attr, 3, 1+CurY, 1+(CurX*3), 0 );

        switch ( ch ) {          /* ----------- process the input -- */
         case 13:   return( CurChar );                  /* Enter */
```

```
        case 27:   return( MENU_NO_KEY );                 /* Esc */

        case 'q':
        case 'Q':  return( MENU_QUIT );                   /* Quit */

        case '8':
        case -72: CurY--;                                 /* up */
                  if ( CurY < 0 ) CurY = MENU_HI-1;
                  break;
        case '2':
        case -80: CurY++;                                 /* down */
                  if ( CurY >= MENU_HI ) CurY = 0;
                  break;
        case '4':
        case -75: CurX--;                                 /* left */
                  if ( CurX < 0 ) CurX = MENU_WIDE-1;
                  break;
        case '6':
        case -77: CurX++;                                 /* right */
                  if ( CurX >= MENU_WIDE ) CurX = 0;
                  break;
        case '7':
        case -71: CurX=0; CurY=0;                         /* Home */
                  break;
        case '1':
        case -79: CurX=MENU_WIDE-1; CurY=MENU_HI-1;       /* End */
                  break;

       default:   DOSBEEP( 100,100 );                     /* unknown key */
       }
    }
}

/* == This pops up a message to announce installation ============
      VioPopup is used here since we expect CharMenu to be
      executed as a detached process.
*/

InstallMsg( err )
unsigned err;          /* 0 means registered OK, else is error */
{
   unsigned waitflag;
   char attr;
   static char msg[] =
      " CharMenu is now active for the current screen group.\r\n"
      "\r\n"
      " Use Alt-C          for the menu.\r\n"
      "      Alt-Spacebar  to repeat the previous character.\r\n"
      "\r\n"
      " Now press any key...";

   waitflag = 1;                 /* wait -- should always work here */
   VIOPOPUP( &waitflag, 0);      /* get popup control */
   if ( err != 0 )
      sprintf( msg, "Can't install Charmenu.  Error %d\r\n", err );

   VIOWRTTTY( msg, strlen(msg), 0 );
   getch();
   VIOENDPOPUP( 0 );                /* release the popup screen */
}

/* =========== This obtains the ID of the current screen group ==
   The byte at offset 24 in the InfoSeg is the current screen group.
   This function just extracts that one value.
*/

CurGrp()
{

   unsigned infoseg, localseg, cg;
   char far *gdt;

   DOSGETINFOSEG( &infoseg, &localseg );
   gdt = (char far *)( (long)infoseg << 16 );
   return( gdt[24] );
}
```

Listings can be downloaded using PCTECHline, 301/740-8383. Parameters: 2400/1200/300 bps, no parity, 8 data bits, 1 stop bit.

Announcing MicroRAM.™ EMS 4.0 for the Micro Channel.™



You selected the Micro Channel for its performance. Now use it! MicroRAM from Tecmar delivers EMS 4.0—with the performance only hardware can offer. So you can take early delivery of advanced OS/2™-like features while using your favorite, off-the-shelf DOS software applications!

Our advanced engineering gives you up to 8MB of RAM per board, per slot with the option of two serial ports.

MicroRAM 512KB For Just $495. Start off with our MicroRAM 512KB and plug in memory as you need it.

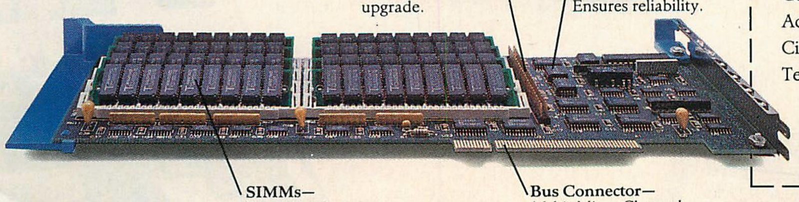
Our standard 512KB and 2MB memory-upgrades snap right onto the board. So you add memory anytime—up to 8MB per board.

OS/2 Compatibility. In addition to DOS, EMS, EEMS and EMS 4.0 compatibility, MicroRAM offers extended memory for protected-mode operating systems like OS/2, XENIX™ and Unix.™

Backed By The Best. MicroRAM is backed by our two-year warranty. And if you have a problem in the first six months, we'll send a replacement within 48 hours—guaranteed!

You made the right decision with PS/2. Now make the right memory and I/O decision. You'll ♥ MicroRAM!

Now call (216) 349-1009 for information. Or, for your free PS/2 I Love You sticker, send the coupon to: Tecmar, Inc., 6225 Cochran Road, Solon, Ohio 44139-3377.



TECMAR

The Power Behind Your PS/2

YES, I want more information on MicroRAM and Tecmar's other PS/2 products!

FREE

Name: ______________________

Title: ______________________

Company: ______________________

Address: ______________________

City: ____________ State: ______ Zip: ______

Telephone: ( )______________________

☐ I'm interested in Tecmar's Special Corporate Demo Discounts!

☐ Send information on Tecmar's full line of products for PCs and ATs.

Send to: Tecmar, Inc., 6225 Cochran Road, Solon, Ohio 44139-3377

CIRCLE NO. 225 ON READER SERVICE CARD

Pixels with Panache

Is it real, or is it the 8514/A? IBM's high-resolution display adapter is not only livening up PS/2 graphics, it is also invigorating the high-end graphics market.

ED McNIERNEY

Along with the PS/2 hardware line announced by IBM in April 1987 came IBM's second attempt at entering the high-performance video market—the Personal System/2 (PS/2) Display Adapter 8514/A and 8514 Color Display. The first attempt, in 1985, was the Professional Graphics Controller (PGC) and Professional Graphics Display, a system that received only moderate support and sales, having been perceived as overpriced and underpowered. Since then, the PC graphics market has matured, in terms of both the performance of the available hardware and in the sophistication and technical demands of users and software developers. IBM's new 8514/A is a wholly different product in a wholly different market, and IBM has evidently learned from the PGC.

Although they work together as a system, the 8514/A adapter and the 8514 display are sold separately. The 8514/A can be used to display more colors in Video Graphics Array (VGA) mode on the PS/2 8512 and 8513 color displays; the 8514 display can be used on any PS/2, without the 8514/A. The 16-inch 8514 display is useful itself, but the 8514/A really makes it shine.

On the whole, the 8514/A is a typical IBM product. It offers no technical innovations or outstanding functionality; all of its features are available from other display adapter manufacturers with either better performance or lower cost. The 8514/A is, however, a robust, general-purpose graphics sys-

PHOTOGRAPHY • HAL RUMMEL

TABLE 1: ***8514/A Video Modes and Display Resolutions***

	ALL SUPPORTED PS/2 DISPLAYS	8514 DISPLAY ONLY
Horizontal resolution	640 pixels	1,024 pixels
Vertical resolution	480 pixels	768 pixels
Scan mode	Noninterlaced	Interlaced
Pixel rate	25.17 MHz	44.90 MHz
Horizontal line rate	31.47 KHz	35.52 KHz
Frame rate	59.94 Hz	43.48 Hz

The 8514/A display adapter can drive any of the PS/2 monitors (the 8514, the color 8512 and 8513, or the gray-scale 8503) at 640-by-80 pixels in either 16 or 256 colors. It also can drive the 8514 color monitor at 1,024-by-768 pixels.

tem that offers reasonable display resolution and color at an IBM price. Perhaps the most significant feature of the 8514/A is that, through its introduction, IBM is helping to publicize the high-end graphics market. Many users who would not buy non-IBM display adapters now find themselves considering the 8514/A, and its popularity will probably increase the size of the growing graphics hardware market.

The 8514/A is a single-board display adapter for PS/2 Models 50, 60, and 80; no version is available for the PC line or PS/2 Models 25 and 30. The adapter is installed in the single Micro Channel slot in each of these machines, which has a small additional video connector; the display monitor connects to a 15-pin D-shell connector on the adapter bracket (see photo 1). Using the adapter with the 8514 display allows presentation of 1,024-by-768 pixel graphics on a 16-inch-diagonal screen (see photo 2). The standard display adapter can display 16 colors simultaneously, while an optional memory expansion daughterboard allows 256 colors. Each color selection is from a palette of 256,000 available colors.

While higher-resolution monitors and display adapters are available, the resolution of the 8514/A and 8514 display represents a reasonable step above the VGA's standard resolution, without excessive monitor costs. In many high-resolution graphics systems, the monitor cost is one-half to two-thirds the price of the entire system; IBM decided to keep the cost of the system down by keeping the resolution at 1,024-by-768 pixels and by using an interlaced monitor rather than the more common noninterlaced type.

In addition to providing enhanced resolution on the 8514 display, the 8514/A can drive any of the PS/2 monitors (the 8514, the color 8512 and 8513, or the gray-scale 8503) at 640-by-480 pixel resolutions in either 16 or 256 colors (see table 1). The 8514/A uses the extended VGA video output from the Micro Channel bus connector to provide perfect VGA software compatibility. (See "VGA: Evolutionary Half-Step," John T. Cockerham, August 1987, p. 74.) When the 8514/A is not running a high-resolution graphics program, it is passive and simply passes the motherboard VGA output through its video connector to the monitor. Because the output display is coming from the VGA rather than from the 8514/A, compatibility is assured.

The 8514/A display adapter continues the PS/2 trend of supporting only display devices with square pixels: the resolutions supported (1,024-by-768 and 640-by-480) maintain a ratio of 4 x pixels for every 3 y pixels. Because standard monitor screens follow this same ratio and are four-thirds as wide as they are high, the resulting pixels come out square. Square pixels make graphics software much easier to develop and can assist in drawing speed. Ever since the Macintosh (the first personal computer to have a square-aspect-ratio display), users have found that such displays offer superior graphics and make it easier to read text.

The 8514/A provides hardware assistance for drawing operations when operating in advanced-function mode. This mode is available for both 640-by-480 and 1,024-by-768 pixel resolutions; it does not require the 8514 display to work. The 1,024-by-768-pixel resolution, however, is available only on the 8514. The display adapter's advanced-function graphics are command driven through a software interface that translates high-level graphics function calls into hardware commands to the display adapter. Because the advanced-function mode must be explicitly started and terminated by the application, the display adapter uses software calls to determine whether or not to operate in VGA display mode.

When the system is powered up, the 8514/A is passive and allows the native VGA graphics to be displayed. Only after the display adapter's software interface (HDILOAD.EXE, a terminate-and-stay-resident, or TSR, utility provided with the adapter) is loaded and an application program makes a call to open it, does the VGA operation cease and the adapter's advanced-function mode go into effect. When the application terminates, it closes the adapter with another call and the VGA operation resumes. Because the 8514/A uses its own local display memory when in advanced-function mode, the original VGA display can be restored after a graphics program ends.

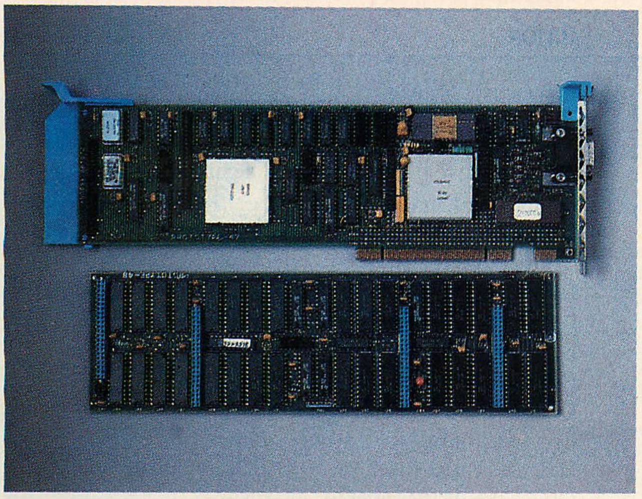
The 8514/A display adapter behaves as a hard-wired graphics processor with a very limited hardware function set that is supplemented by software. In terms of flexibility and function, it probably most closely resembles the NEC 7220—one of the earlier graphics chips. More recent graphics hardware, including the Hitachi HD63484, the Intel 82786, and Texas Instruments' TMS 34010, feature much more flexible and powerful graphics (see "High-performance Graphics: Intel 82786" and "Custom-tailored Graphics: TMS34010," Ed McNierney, July 1987, p. 56 and p. 68 respectively). Although the 8514/A can approach these chips in terms of drawing speed, it falls far short in its limited drawing capabilities. Even the Hitachi chip, the oldest of the group, has hardware support for drawing arcs, a feature the 8514/A lacks.

INTERLACED DISPLAY

The 8514 is the first interlaced monitor that IBM has offered for PCs, although interlaced monitors as a whole are rather common in high-resolution systems. The resolution of a monitor is frequently measured by its horizontal scan rate—the number of horizontal scan lines (the display's vertical resolution) displayed every second. For a 768-line display refreshed at 60 frames per second, a horizontal frequency of about 60 KHz is required.

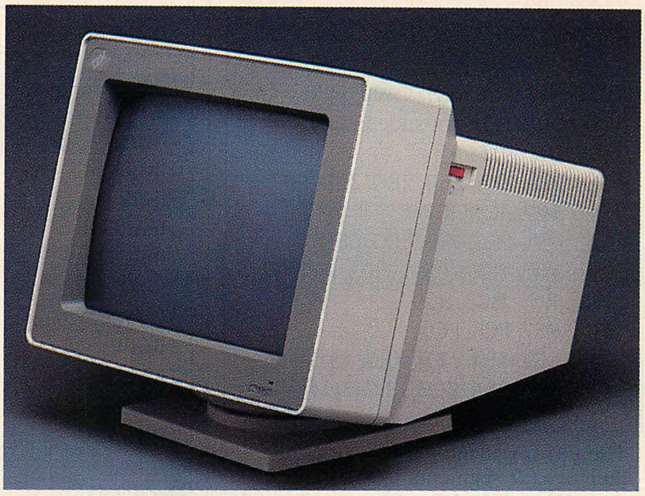
Rather than scanning the entire screen once per frame (about 1/60th of a second), an interlaced monitor displays only every other horizontal scan line during each frame; the alternate lines are filled in on the next scan. The result is that twice as many scan lines can be displayed on a monitor without changing its horizontal frequency. Because the number of lines displayed during each frame can be traded off against the number of times the frame

PHOTO 1: ***8514/A Display Adapter***



The 8514/A adapter board (top) can be used in the PS/2 Models 50, 60, and 80. The display monitor connects to a 15-pin D-shell connector on the adapter bracket. An optional memory expansion kit supports 256 simultaneous colors.

PHOTO 2: ***8514 Color Display***



This display, designed for use with the 8514/A, allows 1,024-by-68-pixel graphics and text displays of 146 characters by 51 lines. This resolution is a reasonable step above the VGA's standard resolution without the excessive costs.

is updated each second, the resolution also can be increased by reducing the frame rate as well.

The 8514 display uses both techniques to adjust from 640-by-480 noninterlaced to 1,024-by-768 interlaced resolution. In 640-by-480 mode, each scan line is displayed on every frame and the entire screen is updated 60 times per second. In theory, interlacing could double the vertical resolution to 960 lines, but slowing updates down to 30 times per second can produce display flicker, an annoying side effect of interlacing. By displaying only 768 lines rather than 960, the 8514 trades off that resolution to increase the frame rate to 44 frames per second.

Combining this scan rate with a longer-persistence phosphor is a good compromise. Even though slow phosphors trade off flicker for smearing, the 8514/A combines the best of interlaced design and slow phosphors, while avoiding their disadvantages. IBM has not solved all problems associated with interlaced designs, but it has found a fertile middle ground. The result is an excellent monitor for imaging, but it still might be unsuitable for CAD, drafting, and technical operations.

Because the perception of smearing or flicker depends on the user and on lighting, a prospective buyer would do well to try an 8514 on site under normal conditions; the display will satisfy most users, but it can be an expensive mistake if it does not.

Like the VGA, the 8514/A uses feedback from the monitor to determine the display hardware. This information controls the start-up operation of the adapter as well as limits access to higher-resolution display modes to only those monitors that can support it. Three of the 15 pins on the monitor connection are used as monitor ID bits and either are grounded by the monitor or left unconnected. By examining these ID bits, the 8514/A can select appropriate behavior for each of the supported displays.

Although the 8514/A routes the VGA video output through its own monitor connector, that output is made available to the standard VGA output connector on the system unit when the 8514/A is operating in advanced-function mode. Such a system is the only supported dual-monitor system for the PS/2 line; an 8514 can be connected to and driven by the 8514/A while a standard 640-by-480-pixel display is connected to the VGA output connector.

INTERFACE INTERACTIONS

Unlike previous IBM display adapters, the 8514/A's hardware control system is not documented—all operations are supported through its 8514/A Adapter Interface software. This restriction allows IBM to modify and upgrade the 8514/A hardware and even to change its memory and I/O addressing without affecting applications that depend on the interface. (See table 2 for a list of the I/O addresses used by the 8514/A display adapter.)

Although the software interface seems good in theory, it has two disadvantages—one for users and one for IBM. Users suffer because the software interface adds another layer between applications and graphics hardware on the adapter. The software interface, because it is general-purpose, cannot be optimized for the particular requirements of an application that may be able to dispense with many of the general features of the interface and gain higher drawing speed.

Here, IBM faces a tough challenge much like the one presented by OS/2's video interface: to make the system expandable and robust, a strict software interface must be followed, but developers must be convinced that the interface provides them with substantially the same performance by writing directly to the video hardware. In the case of OS/2, the character-mode operations available through VIO function calls seem to have satisfied developers; it seems less likely they will be as happy with the 8514/A's software.

The second disadvantage of a software interface is strictly a competitive one for IBM. While hardware interfaces are complex, expensive, or difficult to copy, and easy places in which to hide undocumented features, software interfaces are easy to duplicate and rarely contain undocumented entry points that are of any significant importance to applications software.

Manufacturers of programmable graphics boards using the Texas Instruments 34010 Graphics System Processor have already announced 8514/A emulation support—some new software is uploaded into the display adapter, a new host interface program is written, and software simulation of an

TABLE 2: *8514/A Addresses*

MEMORY LOCATIONS (hexadecimal)
C600:0800 through C700:0FFF
CA00:0000 through CA00:07FF
I/O ADDRESSES (hexadecimal)
100, 101, 102
2E8, 2E9, 2EA, 2EB, 2EC, 2ED
3C6, 3C7, 3C8, 3C9
6E8, 6E9, 6EA, 6EB, 6EC, 6ED
7C6, 7C7, 7C8, 7C9
BC6, BC7, BC8, BC9
FC6, FC7, FC8, FC9
AE8, AE9, AEA, AEB, AEC, AED
EE8, EE9, EEA, EEB, EEC, 2ED

The 8514/A adapter uses blocks of memory (at segments C600H and CA00H) and 43 I/O addresses.

TABLE 3: *Adapter Interface Commands*

COMMAND	ACTION
HLINE	Draws absolute polyline starting at absolute position
HCLINE	Draws absolute polyline starting at current position
HRLINE	Draws relative polyline starting at absolute position
HRCLINE	Draws relative polyline starting at current position
HBAR	Begins filled area
HEAR	Ends filled area
HRECT	Draws filled rectangle
HMRK	Draws a marker symbol at absolute position
HCMRK	Draws a marker symbol at current position
HBBW	Defines Bit-BLT destination at absolute position
HCBBW	Defines Bit-BLT destination at current position
HBBR	Defines Bit-BLT source at absolute position
HBBCHN	Executes Bit-BLT to or from system memory
HBBC	Executes Bit-BLT entirely in display memory
HOPEN	Opens the display adapter interface
HCLOSE	Closes the display adapter interface
HSCP	Moves to absolute location
HQCP	Inquires about current position
HQDFPAL	Inquires about default palette
HINIT	Initializes task-dependent state buffer
HSYNC	Sets adapter to a task-dependent state
HINT	Waits for vertical retrace
HSMODE	Sets display adapter mode
HQMODE	Inquires about display adapter mode
HQMODES	Inquires if display adapter modes are available
HEGS	Clears screen to zero
HSGQ	Sets graphics quality/drawing styles
HSHS	Sets scissor (clipping rectangle)
HLDPAL	Loads palette
HSPAL	Saves palette
HRPAL	Restores palette

8514/A is complete. Many of these programmable graphics boards can offer greater value and flexibility than the 8514/A and will provide serious competition to it. These same board manufacturers would be unwilling to make all the hardware changes required to provide true 8514/A hardware compatibility. Hardware changes increase the cost of every board produced, whereas software has to be written only once. Given these joint pressures on the 8514/A's software interface, it will be interesting to see exactly how long the interface holds up.

Like the standard OS/2 applications program interface (API), the 8514/A uses a function-call mechanism to allow high-level languages to issue commands to the display adapter. (For a discussion of API, see "A Consistent API," Michael Brian Bentley, March 1988, p. 78.) Because DOS does not provide the dynamic linking support available under OS/2, however, an interrupt-based mechanism is required to set up the entry-point linkages.

The 8514/A adapter interface is provided by the HDILOAD.EXE utility bundled with the board. HDILOAD initializes the display adapter, checks its presence and operation, and ties itself into interrupt vector 7FH. The only documented entry point to interrupt 7FH is to obtain a far pointer to an entry-point table; register AX is loaded with the value 0105H and interrupt 7FH is issued. If the call is successful, the carry flag will be clear and CX:DX will form a segment:offset pointer to an entry table. This table consists of 59 segment:offset pointers to the software-command entry points provided by HDILOAD (see table 3).

HDILOAD is called through these entry points, each of which takes as a parameter a far pointer to a parameter block whose contents are specific to that function call. Pascal-style calls are used to access HDILOAD functions. The called routine (the HDILOAD entry point) removes the parameter from the stack and executes a far return. If any values are returned by the function call, they are stored in the structure pointed to by the parameter. The HDILOAD interface software occupies 16KB of system memory when loaded and cannot be uninstalled from the system without rebooting.

The function-call interface to the 8514/A is rather straightforward. Because the adapter's functionality is low-level, the programmer needs to make practically no decisions about how to approach a problem—only one path can be taken. The only complicated steps in an 8514/A application occur at initialization, because the adapter's presence must be tested, the presence of the adapter software interface must be assured, and the proper inquiry and mode setting commands issued.

Most of these operations can be performed by a stock template program that can be written once and reused. Listing 1, 8514DEMO.C, is a sample program that starts up the adapter interface and then displays the string "Hello, 8514/A!" centered on screen, using the standard 12-by-20 pixel font.

PROFICIENT GRAPHICS

The 8514/A provides a high-level graphics interface for line and figure drawing, text output, image loading and manipulation, cursor control, clipping and filling, and inquiries. The adapter interface is not tailored for any particular type of software, and it is unfortunate that the standard primitive set cannot be extended by third-party developers to provide finer control over graphics operations. The primitives provided, however, cover a broad range of applications and supply all the required building blocks for drawing, imaging, and text software.

COMMAND	ACTION
HSLPC	Saves line pattern position
HRLPC	Restores line pattern position
HSBP	Sets bitplane controls (masking, display)
HQCOORD	Inquires about coordinate types
HSCOORD	Sets coordinate types
HESC	Escapes (terminate adapter processing)
HQDPS	Inquires about drawing process state buffer size
HSMARK	Sets current marker shape
HSPATT	Sets current pattern shape
HSPATTO	Sets current pattern origin (alignment) point
HSLT	Sets current line type
HSLW	Sets current line width
HSCOL	Sets current foreground color
HSBCOL	Sets current background color
HSMX	Sets mix (drawing raster operation)
HSCMP	Sets color comparison register
HSCS	Sets character set
HCHST	Draws text string at absolute position
HCCHST	Draws text string at current position
HXLATE	Assigns color index table for text
ABLOCKMFI	Writes alphanumeric character block
ABLOCKCGA	Writes CGA-format alphanumeric character block
AERASE	Erases character cell rectangle
ASCROLL	Scrolls character cell rectangle
ACURSOR	Sets alphanumeric cursor position
ASCUR	Sets alphanumeric cursor shape
ASFONT	Sets alphanumeric character set
AXLATE	Sets alphanumeric attribute color index table

Software interface commands are provided for drawing, fills, and palette manipulation. Commands also control graphics clipping as well as various text operations.

To support a flexible and expandable interface, the 8514/A uses a set of inquiry functions through which the software can determine the state and current mode setting of the installed adapter and adjust its operation accordingly. Because the 8514/A can operate at two display resolutions and two pixel depths, an application should inquire about the current environment settings and then use that information to operate in a resolution- and color-independent manner; such precautions now will reduce or eliminate conversion efforts if IBM chooses to support an enhanced version of this interface on future display adapters.

When the 8514/A is in advanced-function mode, it always operates in an all-points-addressable (APA) graphics mode, although text operations are provided. The three available 1,024-by-768-pixel display modes differ from one another only by the size of the standard text displayed in that mode. The presence of text functions is complementary to the graphics functionality rather than exclusive of it. Three font formats provide text support in resolutions of 85 columns by 38 rows, 128 columns by 54 rows, and 146 columns by 51 rows; all text operations support IBM 3270 text attribute compatibility (except blinking) so that mainframe alphanumeric software can be readily supported on the 8514/A.

Although an OS/2 version of the HDILOAD adapter interface is not yet available, HDILOAD was obviously designed with the new operating system in mind. The HDILOAD interface supports a task-dependent state buffer in which an application's current drawing parameters and mode settings can be stored. The application can inquire about the size of the buffer required to store the state information, allocate that buffer, and then ask that the adapter's state be either saved into that buffer or restored from it.

Because the state buffer is not updated as drawing progresses, it does not interfere with graphics performance. Under DOS, this state-change

information might be used to set up several drawing modes or environments and switch quickly between them—its use is by no means restricted to multitasking systems.

The 8514/A supports a number of sophisticated binary drawing modes, referred to in the documentation as mixes. These modes control the combination of a drawing color with the already existing pixel data and include the simple Boolean operations replace, AND, OR, and XOR (oddly, the NOT raster operation is not supported). Arithmetic operations, including addition, subtraction, averaging, minimum, and maximum are supported, allowing for sophisticated image blending and control. Such operations, when needed, are performed faster in hardware than in software and can be performed faster on the 8514/A than on the VGA.

Drawing operations. On the 8514/A, drawing operations are performed on a pixel-based integer coordinate space—no support for world coordinate systems or floating-point addresses is provided. All absolute coordinates are 32-bit signed integers, although the actual addressable resolution is much smaller. Relative coordinates, available in alternative forms of drawing instructions, are 8-bit signed values.

Drawing is performed in a coordinate space, which spans an x range of -512 to 1,535 and a y range of -512 to 767; in 640-by-480-pixel mode, the corresponding y range is reduced to -256 to 767. The drawing area is larger than the actual displayable area so that drawing objects such as rectangles or wide lines can extend partially beyond the screen area without wrapping around onto the opposite side of the visible display buffer.

The advanced-function interface supplies the standard line-drawing operations, including programmable patterns and widths being used. Eight standard line patterns are available, and a user-defined pattern of up to 48 pixels long is supported. Unfortunately, the line-width support offered is very limited: only lines that are 1 or 3 pixels wide are supported, and no line-end or line-join styles are provided.

Wide-line drawing is accomplished by drawing two new lines adjacent to the original line. There is no guarantee that these three lines will not overlap, so this method of drawing wide lines is useless for anything but overpainting operations. Other operations (such as exclusive-OR) will suffer from interactions among the three lines. As a result, the adapter's wide-line capabilities are of no practical use to software developers. No circle, arc, or other conic section drawing primitives are provided, and arcs must be produced either through the very slow but accurate process of pixel-by-pixel drawing or through the faster but inferior approximation by line segments.

Fills and bit-BLTs. Although explicit polygon filling is not supported, polygonal areas can be defined and filled using the adapter's **begin filled area** and **end filled area** commands. Between these two commands, any closed figures that are constructed with line, rectangle, and move commands are filled with the current fill pattern.

The supported fill patterns are user-defined and can range from 1-by-1 pixels to a maximum of 32-by-32 pixels. Each pattern can be either monochrome or colored; monochrome patterns are expanded using the current foreground and background colors before being applied to the screen data, while colored patterns (up to 8 bits per pixel) are applied with no translation. For solid fills of rectangular regions, a special version of the rectangle command can perform a faster fill.

Bit-BLT operations (image block copies and moves) are supported either for moving graphics images from one place in display memory to another or for transferring images between display and system memory. The command structure for bit-BLT operations is oddly complicated—one of several commands selecting either system-to-display or display-to-system transfers must be executed first, followed by a second command to supply the system memory address of data buffer to be read from or written to. This second command can be replaced by another command that transfers the data to another location in display memory—the 8514/A checks for overlapping source and destination rectangles and adjusts its operation accordingly.

Bit-BLT sources can be either full color or monochrome images. Monochrome images are expanded to full color using the current foreground and background colors in effect when the bit-BLT is executed. The current mix operations are used to allow the bit-BLT data to be combined with the destination data when the destination is in display memory; reads from display memory into system memory are not affected by the mix settings.

Palette control. The 8514/A provides a standard 18-bit (256,000-color) palette that is independent of the motherboard palette the VGA display system uses.

When the adapter is operating in VGA mode, all BIOS palette settings are duplicated on the 8514/A as well as in the VGA palette so that the display output of the 8514/A remains correct. Once the adapter is switched into advanced-function mode, this operation no longer occurs and the adapter palette must be set explicitly by the HDILOAD palette interface commands. When the adapter returns to VGA display mode, the VGA palettes retain the same values they had when advanced-function mode began.

In mode-setting operations, the palette is properly considered to be a global resource that is not part of a task's task-dependent state information. If an application in a multitasking environment does not wish to accept a palette setting that may have been chosen for it by the user or by a palette control program, it can save and restore its palette upon state switching.

Text operations. Text on the 8514/A is divided into two areas: full-function graphics text and alphanumeric text. Graphics text can be defined either as bitmapped or vector characters and can be drawn at any position on the screen. The bitmapped text characters can be either monochrome or color and are combined with the screen data in a manner similar to pattern fills. Proportional spacing is also supported for bitmapped fonts by allowing each character's left and right margin spacing (space added to or subtracted from the standard cell spacing) to be set through a table specification.

Characters cannot be rotated, sheared, or otherwise transformed except through the definition of multiple bitmap or vector fonts. Although rotation of bitmap fonts is difficult and should not be expected from an adapter of the 8514/A's class, stroke fonts usually are defined for no other reason than that they can be rotated to arbitrary angles or scaled to any size. The 8514/A's omission of character scaling or rotation renders stroke fonts impractical. For all but very large fonts, the character definitions would be smaller and the resulting characters drawn faster if the characters were defined as bitmaps instead.

Alphanumeric text is cell-oriented and is supplied to help simulate character-mode displays or to support programs that require rapid display of monospaced text screens. Up to four 256-character fonts can be active at one time, and each character displayed can be from a different font. Character data can be written only to the screen and not read back, because that character data could be mixed with other screen graphics and become unreadable.

Character data are supplied in either 2-byte CGA format (character/attribute pairs) or in 4-byte 3270 format. The 4-byte format allows for independent font selection and more control over character rendering styles, including transparency and color. Font storage for both formats is in PC memory and must be managed by the program. Font display is strictly a drawing operation, and the 8514/A retains no information about the character displayed. If the font colors or font definitions themselves are changed, the text display on the screen will not be updated unless the application explicitly clears the text display and redraws it.

Miscellaneous operations. The 8514/A supplies a number of other graphics functions that assist high-performance graphics applications. A hardware "scissor" or clipping rectangle allows the output of graphics operations to be clipped rapidly to some subset of the display screen. This scissor operates at a hardware post-clipping level: every pixel of the figure is generated and then drawn. If the pixel does not lie within the scissor rectangle, the drawing operation has no effect. The interface has no facility for rapid preclipping of drawing operations, so if an application intends to work with a very small scissor rectangle, it would be better to examine and preclip its drawing operations rather than leaving all clipping to the hardware scissor.

In 640-by-480, 16-color display modes, the 8514/A has sufficient memory to support two full pages of display. Drawing can occur in either page while the selected page is visible, and instantaneous toggling between display pages is provided. This multiple-page support is provided through the adapter's bitplane enable and display mechanisms: any set of the available bitplanes may be write-protected to prevent them from being modified by drawing commands, and any set of bitplanes can be blanked from the display without corrupting their contents.

The 8514/A lends itself to applications that use palette and plane masking to achieve visual effects. An application can load an 8-bit screen display with data defining a different image in each bitplane and then use plane masking to switch between eight full-screen, single-color displays. Although using eight images is not common, some applications may wish to use a standard 128-color display using seven bitplanes

and keep the eighth plane constantly updated with a context-sensitive help screen. By enabling output from the eighth plane, the help screen can be made to appear instantly overlaying the graphics display beneath it. When the help message is no longer needed, it can be removed just as fast without damaging the working display.

PERFORMANCE REVIEW

The 8514/A system is a difficult one to benchmark, because its performance is the result of a complex set of interactions between the PC software interface and the on-board hardware. Because the hardware interface is not published, it is difficult to tell which operations are performed by the adapter and which are synthesized by the interface software. By providing comparative measurements on both the 10-MHz PS/2 Model 60 and the 20-MHz PS/2 Model 80, some idea of the board's performance can be obtained independent of the software interface. The interface is, however, an essential portion of an application's overhead in using the 8514/A and should be considered when planning performance estimates.

Drawing speed. The benchmarks contained in 8514TEST.C (which is available on PCTECHline) are designed more as examination and research tools than as stand-alone performance measurements, because they test only selected but common adapter operations rather than attempting to cover all supported graphics primitives. Each benchmark performs the same set of graphics operations under two different conditions: execution in a tight loop as quickly as possible, and one command at a time with a delay before each command is issued. In addition, each pair of tests was run on both the PS/2 Model 60 and the Model 80. The results not only indicate the performance of the board under varying conditions, but also help reveal some of the interactions between the hardware and software components of the system.

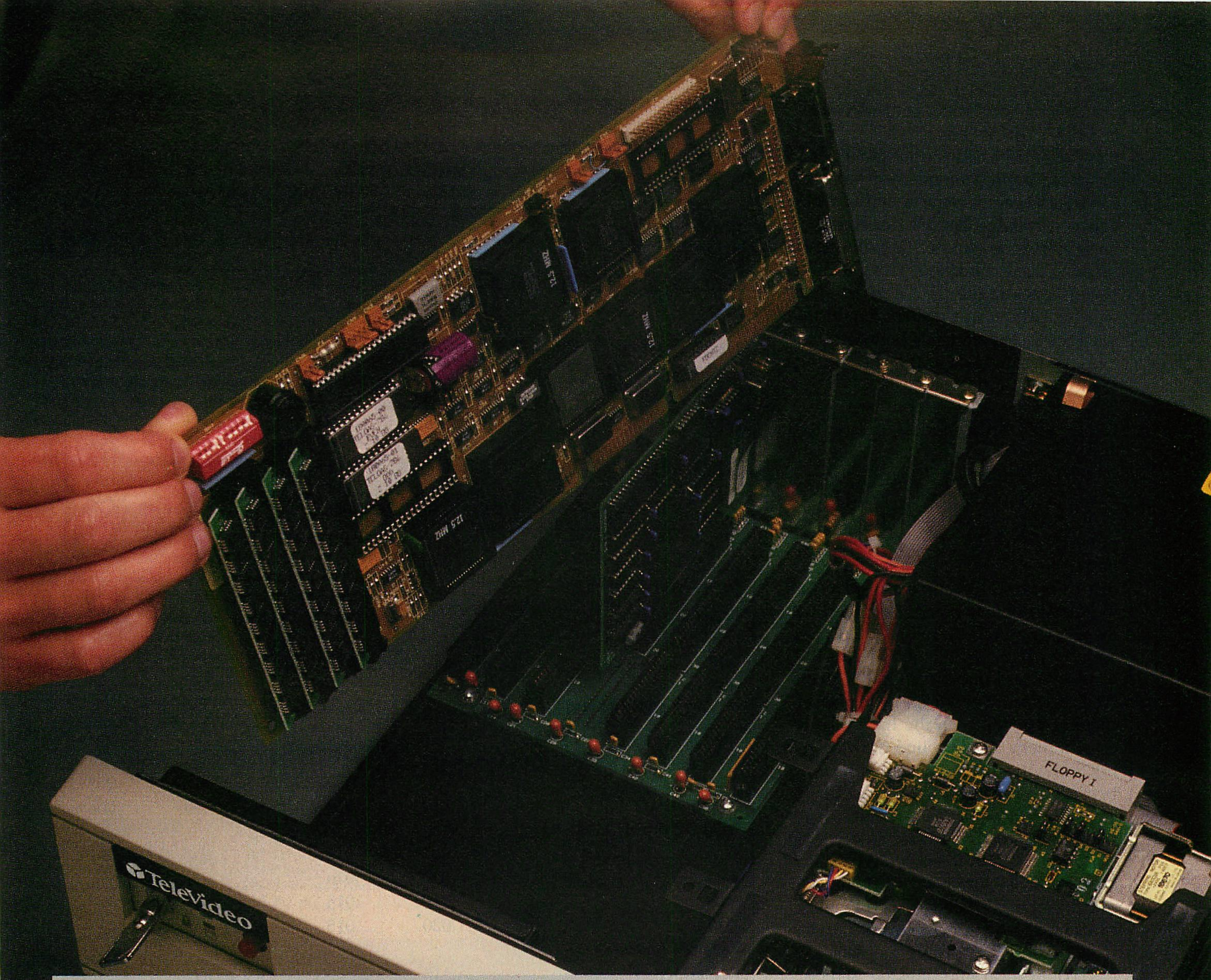
For operations in which the display adapter interface must perform significant preprocessing, issue a command, and then return while the command is being executed, a noticeable difference in performance will appear between the two types of tests. If a large number of such commands are executed in rapid succession, significant overlap will result, during which the 8514/A adapter will be executing the first command and the PS/2 will be preparing for the second command. This parallelism should show up as a performance increase when a command is issued in a tight loop rather than in single-shot situations.

The benchmark tests include the following operations, each of which is executed in replace, XOR, add, and average mix modes:

- Horizontal lines—fills the screen by drawing full-width horizontal scan lines. This should measure the maximum line drawing rate.
- Vertical lines—fills screen by drawing full-height vertical scan lines. These lines may be as slow as random lines, but they can be optimized in hardware. If no difference exists between vertical and random lines, it is assumed no optimization exists.
- Screen clear—clears the screen to a solid color. It represents the maximum rate at which the adapter can update the display screen.
- Random long lines—draws 10,000 random lines, each 768 pixels long. These lines measure optimum random line performance and can be compared with vertical line drawing.
- Random medium lines—draws 10,000 random lines, each 40 pixels long. The 40-pixel vector is a typical line length for drafting and technical drawing applications. This test most closely simulates real-world application requirements.
- Random short lines—draws 10,000 random lines, each 4 pixels long. This test emphasizes the overhead of the adapter interface, because the line drawing takes very little time.
- Random solid-filled rectangles—fills 1,000 rectangles with a solid color. These rectangles should show the maximum pixel output rate.

In order to most closely simulate the 8514/A benchmark test on a VGA system, the VGALOAD.COM interface was developed. This interface provides a software function set that exactly matches a small subset of the 8514/A's software interface as provided in HDILOAD. By implementing the functions used by the benchmark software in VGALOAD.COM, the same benchmark program could be run each time, allowing a head-to-head comparison of the two adapters. (The C-language source and executable versions of VGALOAD and 8514TEST can be downloaded from PCTECHline.)

The drawing software in VGALOAD is reasonably optimized for the tasks being presented. Different performance could result if the software were expanded to implement the full 8514/A feature set or tailored to the requirements of a specific application.



We've Opened Up All Kinds Of Possibilities.

TelOAS™ from TeleVideo.® Open Architecture Solutions™ that can be anything you want them to be.

The new TelOAS solutions are entirely new ways of looking at TeleVideo computers. Very flexible systems that give you the choices you need to build the system you need now, then easily expand and upgrade later.

Choose the cabinet with the proper number of slots you need: 4, 8 or 12. Choose the CPU you want: 8088, 80286 or 80386. Choose the drives you want, the monitor you want, the keyboard, the footprint and the price you want. Choose the software to match your needs: MS-DOS,™ MS-OS/2,™ UNIX® System V.3, NetWare,™ and others. With the TelOAS Systems, the choices are all yours.

The TelOAS/I cabinet has a passive back plane with 4 board slots. It accepts TelOAS 8088 or 80286 boards in a low-profile, small-footprint case that fits anywhere.

The TelOAS/II cabinet has 8 board slots, room for 4 half-height drives, and accepts 80286 or 80386 boards. With the right peripherals, you have an affordable small-footprint workstation that runs both MS-DOS and UNIX System V.3.

The TelOAS/III cabinet has 12 board slots, room for 2 full-height and 2 half-height drives, and accepts 80286 or 80386 boards. With the right peripherals and software, you get a powerful 386 engineering workstation, a LAN fileserver, or a multi-user host system.

The TelOAS Open Architecture Solutions use the most advanced surface-mount technology to reduce the number of components and make the systems easier to service. The diagnostic features give you extensive tests of hardware. Together with our TeleVideo one-year limited ON-SITE warranty,* you're sure of getting quality products that give you the most for your money.

For more information on the new Open Architecture Solutions that can be anything you want them to be, call toll-free, or write today.

TeleVideo Systems, Inc.
1170 Morse Avenue, Sunnyvale, CA
94088-3568.

TeleVideo®
THE VISION YOU NEED TO SUCCEED.

Call 1-800-835-3228

CIRCLE NO. 143 ON READER SERVICE CARD

Analysis of the benchmark results in table 4 reveals that the 8514/A display adapter provides noticeable performance improvements over the standard VGA adapter—the speedup varies between twice as fast for horizontal line drawing to almost 100 times as fast for long vector drawing. The general performance of the 8514/A varies quite widely, and the test results demonstrate some of the interactions between the 8514/A's hardware and the software interface that supports it.

All of the drawing tests were run using four writing modes: replace, exclusive-or, add, and average. Under theoretically optimal conditions, performance for replace mode should be about twice as good as for that of the other three modes. This difference is due to the fact that the other three modes require the current display data to be read before being combined with the drawing data; therefore, each operation requires two memory accesses (one read and one write) rather than the single write required by replace mode. Depending on the hardware environment, random line drawing may not show this much of a difference, because replacing only one pixel in a byte of memory still can require that the byte be read, modified, and rewritten. An exception to this rule can be found in a situations where memory organization is designed such that masking can be performed on the write without prereading the data.

For vertical line drawing, screen clearing, random rectangles, and long random vectors, the 8514/A shows better performance in replace mode than in the other three modes. This observation is consistent with the memory access times required for each operation; the replace mode is not twice as fast as the others because the general overhead of the drawing operation is still a significant fraction of the time required. For medium and short lines, however, all four writing modes show the same performance; in addition, performance for medium lines is about 10 times that for short lines. Examining the results shows that the number of lines drawn per second stays about the same, and since short lines are only one-tenth as long as medium lines, the number of pixels per second goes down accordingly.

The linearity across writing modes and the linear reduction in pixel rate as the line length goes down indicate that the hardware is not being used fully. In fact, for most vector drawing situations, the speed of the software interface is the limiting factor for the system's performance, and the system requires almost the same amount of time to draw any line, regardless of its length. This conclusion also is supported by the fact that medium and short lines (and long lines to a lesser degree) are drawn twice as fast on the Model 80 as on the Model 60. Because the 8514/A hardware is not running any differently, this speedup must be attributed to the execution speed of the software interface. It is, however, unfortunate to discover that the software interface is such a hindrance to vector drawing that even the 20-MHz Model 80 cannot run it fast enough.

VGA results behave as expected for an all-software interface. One exception is the variation in exclusive-or drawing for horizontal lines, screen clear, and random rectangles. Because the VGA has hardware support for exclusive-or and replace modes (but not for add and average), these modes show similar performance under memory-intensive operations. Line drawing does not show such a difference because the speed is dominated by the software line-draw algorithm rather than memory-access time.

At its best, the 8514/A performs well, showing peak drawing rates of over 25 million pixels per second for

TABLE 4: ***Benchmark Results***

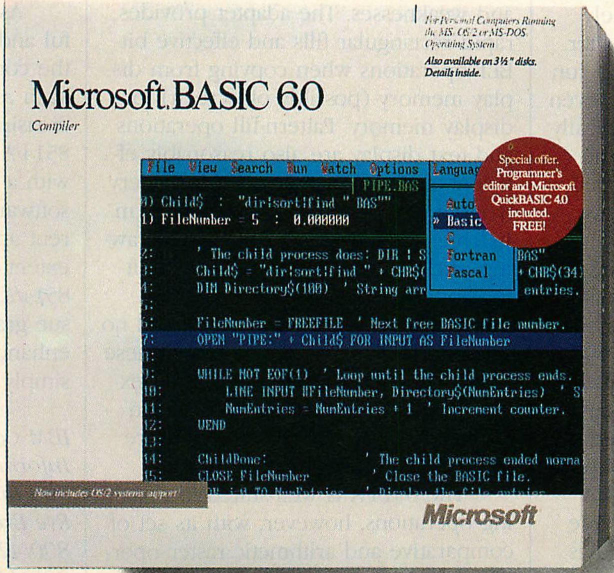
	10-MHz MODEL 60		20-MHz MODEL 80	
	8514/A	VGA	8514/A	VGA
HORIZONTAL LINES				
Replace	6.631	3.024	6.659	3.802
XOR	6.598	3.042	6.598	3.779
Add	6.631	1.427	6.598	1.865
Average	6.598	1.416	6.598	1.845
VERTICAL LINES				
Replace	4.206	.163	4.345	.258
XOR	2.604	.213	2.562	.258
Add	2.553	.158	2.553	.258
Average	2.562	.146	2.553	.257
SCREEN CLEAR				
Replace	25.534	3.213	25.700	4.000
XOR	13.284	3.213	12.977	3.938
Add	13.020	1.471	13.240	1.903
Average	13.020	1.456	13.284	1.875
RANDOM LONG LINES				
Replace	4.062	.042	5.210	.090
XOR	3.474	.041	3.688	.083
Add	3.727	.039	3.960	.079
Average	3.360	.039	3.566	.080
RANDOM MEDIUM LINES				
Replace	.241	.035	.503	.075
XOR	.240	.035	.506	.072
Add	.231	.034	.478	.067
Average	.238	.033	.494	.068
RANDOM SHORT LINES				
Replace	.022	.014	.045	.031
XOR	.020	.014	.042	.030
Add	.019	.013	.040	.027
Average	.020	.015	.043	.032
RANDOM RECTANGLES				
Replace	25.413	2.359	25.413	3.245
XOR	12.964	2.477	12.866	3.313
Add	12.865	1.249	12.865	1.701
Average	12.924	1.263	12.924	1.711

All results are in millions of pixels per second.

The Micro-Channel-compatible 8514/A display adapter provides significant performance improvements over the standard VGA adapter, especially in instances such as screen clearing and rectangle generation, in which a rather small number of software commands generate a large number of pixels on the screen.

NEW!

For pros. No cons.



Most BASICs are designed for people who only use BASIC part of the time.

Finally there's a new BASIC designed for professionals who spend their life with it. It's Microsoft® BASIC Compiler 6.0. It's got all the tools you need to make writing terrific code easier. Whether you want the implementation flexibility of MS-DOS®. Or the advanced capabilities of OS/2.

With OS/2, you can develop large, sophisticated applications that go beyond the 640K barrier. Applications that take advantage of up to 16MB of RAM. And that exploit the potential of today's microprocessors.

Microsoft BASIC Compiler 6.0 has features you can work with, not around. Runtime library control helps you write the most compact manageable code. The customizable runtime library lets you add subroutines that can save disk space.

Since speed and hardware requirements vary, you can choose the floating-point math that suits you from among three math libraries. Including 8087/80287 coprocessor support, a floating point emulator and a fast math library for systems without math co-processors.

For the first time ever in a BASIC, you can get our popular advanced debugger, CodeView.® Now you can sort out program logic at the source code level. Even if you happen to be using more than one language.

And to get your DOS projects off to a quick start, Microsoft BASIC Compiler 6.0 includes Microsoft QuickBasic 4.0.

Call (800) 541-1261, Dept. D86, and we'll give you the name of your nearest Microsoft dealer. Because at 6.0, BASIC has finally come of age.

OS/2 Features
- Contains all of the OS/2 utilities to make protected-mode as well as real mode applications.
- OS/2 language additions such as SHELL, OPEN PIPE, ON SIGNAL and SLEEP.
- First programmer's editor that lets you write both OS/2 and MS-DOS programs.

CodeView Features
- Trace execution, set breakpoints, set watch variables, view source code, see data automatically updated as variables are displayed.

BASIC language features
- Fastest stand alone, multiple module executable with no runtime fees.
- User defined event trapping.
- Multiple module error handling.
- Supports records, recursion, huge arrays.
- Additional data types—4 byte integers, fixed length strings.
- Supports all major structured programming constructs including SELECT CASE and FUNCTION procedures.

Microsoft®

screen clears. At these higher rates, the hardware does all the work of the system and shows that, for graphics performance, a 10-MHz Model 60 can keep up with IBM's top-of-the-line hardware. For many operations, the 8514/A gives a greater relative speed improvement in the Model 60 because the Model 80's VGA is software-dependent on a much faster processor.

Because the 8514/A's only published interface is to the HDILOAD primitives, this interface insulates the developer from changes in the hardware, but it also prohibits extremely high-performance graphics. No matter what the application does, it has to run through a software interface layer, even if it is willing to be adapted specifically for that hardware. Unfortunately, the price paid for this insulation is speed, and in many of the 8514/A's primitive operations that price is high.

In fact, research on controlling the 8514/A through direct hardware manipulation indicates that line drawing and text output could be speeded up 500 percent by eliminating the current software interface and substituting a customized one as part of an application's device-driver support. IBM should publish the specifications for the 8514/A hardware interface or at least provide efficient OS/2 and Microsoft Windows device drivers to replace the present software interface.

PLUSES AND MINUSES

On the whole, the 8514/A hardware seems to be a solid, well-designed adapter with performance approaching that of other graphics systems. It is therefore doubly unfortunate that this high-quality hardware is crippled by IBM's refusal to document the board's hardware interface and insistence upon a software interface layer that severely hampers the board's performance. Although typical graphics applications that use the adapter software interface still will perform better than they would on a VGA, the 8514/A will not really be at its best until direct hardware support is available.

Applications written to the 8514/A should carefully consider its strengths and weaknesses. The adapter provides rapid rectangular fills and effective bit-BLT operations when copying from display memory (possibly off-screen) to display memory. Pattern-fill operations and text display are also reasonably efficient, and the adapter provides a very flexible plane mask and palette system. Its chief weaknesses are in figure drawing, such as required in CAD or drafting applications—it supports no arc primitives, no useful wide lines, and no line end and join styles. Although these primitives can be synthesized from existing ones, the overhead incurred in repeated calls to the adapter software interface will hinder performance.

The 8514/A is well-suited to imaging operations, however, with its set of comparative and arithmetic raster operations and effective palette control. Image display and paint applications, full-color page preview, or solid modeling applications can use the 8514/A's capabilities. Although virtually no applications are available for the 8514/A, look for the first compelling applications to appear soon.

The 8514/A indicates IBM's renewed interest in high-resolution PC graphics. Although the adapter is by no means the fastest, or most colorful, or highest resolution adapter available, it provides a reasonable set of primitives that allow effective use of the adapter in typical PC graphics applications. If nothing else, the visibility of this adapter will help enhance current awareness of, and interest in, high-resolution adapters and help stimulate the development of applications that depend on it.

As user demand for more powerful and sophisticated graphics increase, the contributions made by products such as the 8514/A will become increasingly important. Fortunately, the 8514/A has gotten off to a good start with a reasonably efficient extensible software interface that can support current application demands to a large extent. If IBM succeeds with the 8514/A, it can use that success to pursue graphics technology to extend and enhance the state of the art rather than simply catching up to it.

IBM Corporation
Information Systems Group
900 King Street
Rye Brook, NY 10573
800/426-2468; for nearest dealer, 800/447-4700
PS/2 Display Adapter 8514/A, $1,290
8514 Color Display, $1,550

CIRCLE 336 ON READER SERVICE CARD

Ed McNierney is a principal engineer at Lotus Development Corporation. His focus is on developing graphics hardware interfaces.

LISTING 1: 8514DEMO.C

```
/* 8514/A Demo */
/* Requires ibmafi.h, callafi.obj, hdiload.exe, and stan1220.fnt    */
#include <dos.h>                /* DOS Memory Management interface  */
#include <io.h>                 /* C Low-Level File I/O interface   */
#include <fcntl.h>              /* C File Control definitions       */
#include <malloc.h>             /* C Memory Allocation interface    */
#include <string.h>             /* C String Handling interface      */
#include "ibmafi.h"             /* IBM Display Adapter interface    */
#define  FONTFILE_NAME "STAN1220.FNT"
#define  COLOR_WHITE    3
#define  MESSAGE       "Hello, Display Adapter 8514/A!"
#define  MAX_STR        40
HOPEN_DATA  open_data   = { 3, 0, 0 };
HCLOSE_DATA close_data  = { 2, 0 };
HSCS_DATA   font        = { 4, 0 };
/*  Local Function Prototypes */
struct  CharSetDef  *LoadFont (char *);
int                 main (void);
/*  Local Functions */
struct CharSetDef *LoadFont(fileName)
char *fileName;                 /* Name of the font file to use     */
    {
    word fontlen;
    struct FontFileDefn *ioaddr;
    int f_id;
    struct CharSetDef *a_csd;
    /*
       Try to open the specified font file.  If it can't be opened,
       return an error.
    */
    if ((f_id = open (fileName, O_RDONLY | O_BINARY)) == -1)
        return NULL;
    /*
        Find the size of the font file, allocate a buffer, and load
        the file into it; return an error if buffer can't be allocated.
    */
    fontlen = (int) lseek (f_id, 0L, 2);
    if (!(ioaddr = (struct FontFileDefn *) malloc (fontlen)))
        return NULL;
    lseek (f_id, 0L, 0);        /* back to start of file            */
    read (f_id, (char *) ioaddr, fontlen);
    /*
        Load up the character set definition tables with the address
        at which the font was loaded.
    */
    a_csd = (struct CharSetDef *) (((char *) ioaddr) +
            ioaddr->page_array[ioaddr->def_page].csd_offset);
    a_csd->chardef1 = ((char far *) ioaddr) + ((long) a_csd->chardef1);
    a_csd->chardef2 = ((char far *) ioaddr) + ((long) a_csd->chardef2);
    a_csd->chardef3 = ((char far *) ioaddr) + ((long) a_csd->chardef3);
    a_csd->indextbl = (int far *)
              (((byte far *) ioaddr) + ((long) a_csd->indextbl));
    a_csd->enveltbl = ((char far *) ioaddr) + ((long) a_csd->enveltbl);
    /*
       Close up and return the loaded address.
    */
    close (f_id);
    return a_csd;
    }
int     main ()
    {
    HINIT_DATA          taskState;
    HQDPS_DATA          stateBufferInfo;
    HQMODE_DATA         modeData;
    HSCOL_DATA          textColor;
    HCHST_DATA(MAX_STR) drawString;
    int                 retVal;
    int                 messageHeight, messageWidth;
    union REGS          inregs, outregs;
    struct SREGS        segregs;
    /*
       Check to see that the adapter interface is installed.  If it is
       not, we can't proceed.
    */
    if (getafi () == NULL)
       {
       printf ("Please run HDILOAD to install the Adapter Interface.\n");
       return 1;
       }
```

```
/*
    The adapter interface is installed, so the entry points in it
    can be called safely.  Open the adapter to its default state
    and verify that the open worked.
*/
HOPEN (&open_data);          /* Open the Adapter Interface         */
if (open_data.iflags)        /* If an error occurred on the open */
    {
    printf ("Adapter Open error: %X\n", open_data.iflags);
    return 1;
    }
/*
    Allocate a block of memory for the task-dependent state buffer.
    This buffer is filled in by a call to HINIT, and must be set up
    even if the adapter's state will not be switched later.
*/
stateBufferInfo.length = 6; /* Size of the returned buffer info */
HQDPS (&stateBufferInfo);
/*
    The state buffer info block now holds the size of the task-
    dependent state buffer required by HINIT.  Since HINIT is passed
    a segment value as a pointer to the buffer and assumes that the
    buffer begins at offset 0 in that segment, we'll call the DOS
    memory allocation service to allocate an appropriate block.
*/
inregs.h.ah = 0x48;          /* Allocate DOS memory segment       */
inregs.x.bx = stateBufferInfo.size;
intdos (&inregs, &outregs);
taskState.segment = outregs.x.ax;
taskState.length = 2;        /* Size of the segment pointer       */
HINIT (&taskState);          /* Initialize the opened adapter     */
modeData.length = 18;        /* Get the adapter's default mode    */
HQMODE (&modeData);
/*
    Now we're all set to go.  Everything up to here has been
    canonical code - now we get to the real application.  Load the
    font file and display a string on the monitor.  The font file
    used is the standard 12 x 20 font file supplied with the
    adapter interface code.
*/
if (!(font.address = LoadFont (FONTFILE_NAME)))
    {
    printf ("Font file %s could not be loaded!\n", FONTFILE_NAME);
    retVal = 1;
    }
else
    {
    textColor.length = 4;
    textColor.index = COLOR_WHITE;
    HSCOL (&textColor);      /* Set drawing color to plain white */
    font.length = 4;
    HSCS (&font);            /* Set the character set to be used */
    /*
        Calculate the length and height of the message in the current
        font so that we can center it properly.
    */
    messageHeight = font.address->cellheight;
    messageWidth = font.address->cellwidth * strlen (MESSAGE);
    /*
        Draw the string properly centered on the screen.
    */
    drawString.coord.x_coord = (modeData.width - messageWidth) / 2;
    drawString.coord.y_coord = (modeData.height - messageHeight) / 2;
    drawString.length = sizeof (coord_pr) + strlen (MESSAGE);
    strcpy (drawString.string,MESSAGE);
    HCHST (&drawString);
    retVal = 0;
    }
getch ();                    /* Let the reader admire our work    */
/*
    Release the task state buffer and close the adapter.
*/
inregs.h.ah = 0x49;          /* Free DOS memory segment           */
segregs.es = taskState.segment;
int86x (0x21, &inregs, &outregs, &segregs);
HCLOSE (&close_data);        /* Close Adapter Interface           */
return 0;                    /* Successful operation              */
}
```

Listings can be downloaded using PCTECHline, 301/740-8383. Parameters: 2400/1200/300 bps, no parity, 8 data bits, 1 stop bit.

E-Mail Arrives Part 2

The three leading third-party E-mail implementations—Higgins Mail, The Network Courier, and cc:Mail—offer dependable local and wide area mail service. Their features, however, vary widely from product to product.

STEVEN S. KING

The LAN E-mail industry is prospering. By 1987, some form of E-mail had been installed on networks in many major corporations, government institutions, universities, and smaller sites. The trend is continuing and picking up speed in 1988 as developers realize that E-mail has universal appeal. This bodes well for those considering the implementation of an E-mail system.

E-mail is often the province of third-party software packages that run on any of the major LANs. Three of the leading products of this type, reviewed here, are Higgins Mail from Conetic Systems, The Network Courier from Consumers Software, and cc:Mail from PCC Systems. All of these products support dependable local and wide area E-mail systems. However, their functions, such as routing and administration, vary enough to affect their suitability for different applications.

These E-mail functions were described in detail in part 1 of this article (April 1988, p. 106), along with a number of design approaches such as peer-to-peer and central-branch configurations. Part 1 concluded that dependable LAN-based E-mail systems are deployable now—and have many benefits for interpersonal communication and data transfer, particularly for organizations with LANs already in place.

The next year or two should see significant new entries in the market for wide area E-mail. Ashton-Tate has announced a product based on Action Technology's Message Handling System, designed to process native application file formats and transmit them via LAN-based store-and-forward networks. Microsoft officials say their company is developing a mail package for LAN systems. Both companies see LAN E-mail as a central element in their group-productivity software offerings.

The obvious advantage of an E-mail product from a company such as Microsoft is the potential for seamless integration with operating systems and applications. OS/2's multitasking and interprocess communications facilities are well suited to E-mail applications, which could benefit by running as a coresident process. Under OS/2, a word processor could create a file to be sent to another network user, and then call the E-mail program waiting in the background to transport the file in its native format. With this approach, the E-mail software can run as a low-priority background process, sending and receiving mail while the user works with other applications in the foreground. The Network Courier provides similar functionality with a terminate-and-stay-resident (TSR) program (under DOS) that allows users to send and review mail without exiting their current applications.

A LAN E-mail system integrated with OS/2 could be a remarkable communications tool. Many developers of high-end E-mail software have expressed interest in OS/2, both for mail servers and end-user mail applications. Tight integration of mail with applications and the operating system are priority considerations—whether looking at current products or anticipating developments in the LAN E-mail market.

A PIECE OF HIGGINS

Higgins Mail was originally the E-mail portion of Higgins, a group-productivity suite from Conetic Software that includes an integrated client database, project tracking, group calendar, expense report, note taker, and other functions. The demand for LAN-based E-mail was strong enough to influence Conetic Systems to unbundle its mail software and sell it separately for $495.

Higgins Mail provides local area capability. Wide area mail capability requires a second product, Higgins Exchange, sold for $1,595. When both Higgins Mail and Higgins Exchange are installed on a file server, full post-office and mail-server functions are available. Remote mail users require yet another $395 product, Higgins Remote, which runs on stand-alone or laptop computers. Higgins Remote can support five mailboxes on a remote PC, allowing multiple users on one site to share a single copy of the product. Oddly, it has the desktop utilities of the original Higgins suite, so remote mail users get the calendar, note taker, and so on, whether they want them or not.

Conetic Software offers a gateway to 3Com's 3+ Mail in the form of Higgins To: 3+, for $495. It exchanges messages between Higgins Mail and all 3+Mail post offices that run on 3Com networks.

Another Conetic product is Higgins Mail API, a development kit for users of Higgins Exchange who wish to develop custom mail gateways. This $995 product includes a library of C functions used to build external delivery programs. The routines export queued mail messages from the Higgins system, convert them into a readable format, and call the user's program to make delivery. Higgins Mail API also converts incoming messages from dissimilar systems into the Higgins internal format. With this system, developers can build gateways between Higgins Mail and other mail systems or network application programs.

The origins of Higgins Mail as part of a larger application puts this product at a slight disadvantage when compared with other E-mail applications. Each Higgins Mail user has an elaborate set of files and DOS directories created in a large Higgins Mail subdirectory system. The remote-product installation process, for example, creates nine different DOS directories and loads more than 100 files. In comparison, the other two products, The Network Courier and cc:Mail, use one or two directories each for program and data files.

Another side effect of Higgins Mail's structure is that attachments are not stored in the central message base, and hence, are not encrypted. Because mail users have access to the mail directories, unencrypted attachments are vulnerable to disclosure on the DOS level—an undesirable situation for systems transferring sensitive attachments. Nonetheless, Higgins Mail has many outstanding features to recommend it.

When initially establishing Higgins Mail accounts, the mail administrator assigns each user an address using

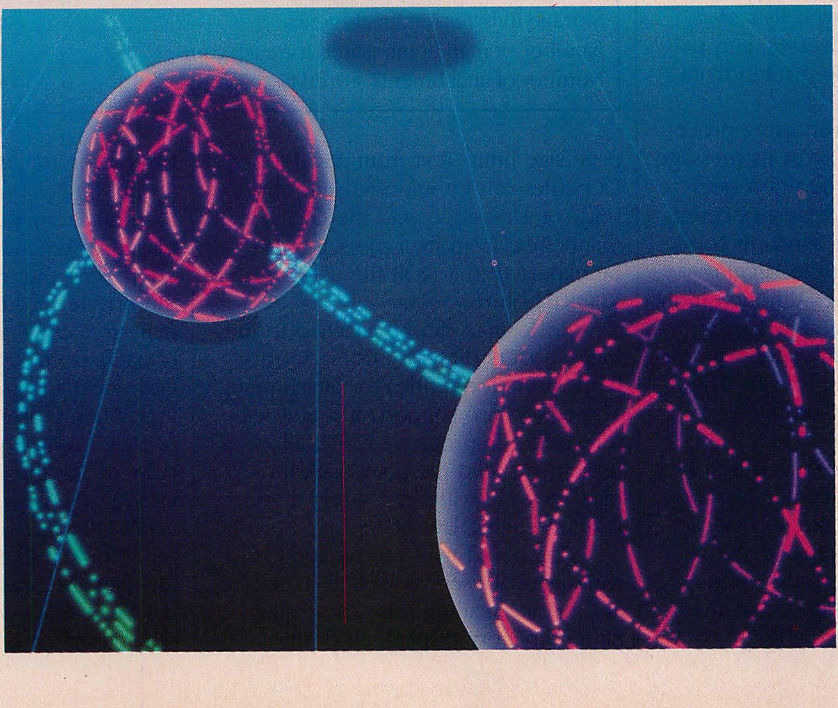
UserId@WorkgroupId@DomainId

UserId is the user's personal identification and unique within the user's workgroup. A workgroup in Higgins Mail is equivalent to a post office; a number of workgroups together are called a *domain*, a group of connected post offices on a single site.

This hierarchical naming scheme is a powerful feature. Administrative and routing functions reflect the hierarchy in their nomenclature and operation. If a user is assigned a valid address, the Higgins Exchange mail servers can locate that user's home domain and workgroup during mail delivery, whether local or remote. One copy of Higgins Exchange, running on a mail server, can support mail transfer for all the post offices on one site (domain).

Routing to remote domains can be direct or via another domain. Multiple hops, or transfers, and indirect routing are supported—making complex wide area mail configurations possible.

With the Higgins Mail package, the mail administrator can set many different parameters for configuring routing tables. The minimum amount of mail that triggers



PHOTOGRAPHY • STEVEN HUNT

a call to a remote post office can be set within three different time periods: day, evening, and night. The minimum amount is a weighted figure that takes into account not only the number of mail items, but also the urgency of each one. The administrator can adjust the values for weighing mail items. The minimum and maximum acceptable periods between calls to a given post office can also be determined by the administrator. Phone numbers or network paths to other post offices in the mail system are entered in routing tables with a menu.

The administrator can also assign usage privileges to each user. One of these privileges specifies send-only or receive-and-send mail activity. Another suppresses a user's name from being listed in the system's public name directories—either local or remote, or both; this parameter is usually assigned to company executives or other persons requiring restricted access.

Another user parameter instructs the mail system to print a copy of incoming mail on the network printer. This is useful for including persons without workstations in the mail system. With the mail-forwarding feature, a user can request mail be temporarily forwarded to another local or remote user. Also, a user's log-in function can be temporarily disabled if, for example, that user is on vacation.

Higgins Mail maintains excellent records of its delivery activities. Statistics available to the mail administrator include number of calls, retries, mail items, urgent items, and bytes transferred, as well as year-to-date figures for all of the preceding numbers. These statistics, along with lists of users, workgroups, addresses, and parameter settings, can be printed from report printing menus.

An end-of-month reporting procedure checks the integrity of its message base, reindexes files when necessary, and resets the monthly statistics counters. One administrative function conspicuously absent from the Higgins package is mail purging. Although individual users can manage and purge their own mail items, the administrator cannot perform a system purge. The implications of this are, of course, a larger message base.

The Higgins Mail editor is a full-screen note taker with cursor-key control, editing functions, and an insert mode. Messages are limited to 120 lines. A search operation can locate strings within the message text—not extremely useful, considering the two-page size limit. Text from ASCII-format DOS files can be imported into the body of a message with the GET command. Text from mail messages can be exported to a DOS file using the XPORT feature.

Enclosures can be attached to messages; a DOS directory listing feature helps users locate files for attachment. The editor also supports a useful feature called In-progress, which allows a user to suspend the composition of a mail message without sending it and resume editing later.

Composed mail can be addressed to individual users as well as to public and private mailing lists. The default name directory displays all of the names for the user's immediate workgroup. Names of mail users in other local and remote workgroups are displayed in separate name directories. Addressing the mail is made easy with the private mailing-list feature. Each mail user can create and maintain lists of users from different workgroups—both local and remote.

When a user receives mail, the ANSWER option permits an immediate response to the message. While still viewing the mail, the user selects ANSWER, enters a reply, and stores the reply for sending. The FORWARD feature lets the user forward mail after it has been read, with comments, to other users or groups of users. Higgins

TABLE 1: ***E-mail Features Comparison***

	CONETIC SYSTEMS	**CONSUMERS SOFTWARE**	**PCC SYSTEMS**
PRODUCT	Higgins Mail	Network Courier	cc:Mail
VERSION TESTED	1.1	1.08	1.2
BASE PRICE	$495	$695	$595
Number of users in base configuration	20	100	25
With wide-area capabilities	$1,595	$995	$1,295
ROUTING AND COMMUNICATIONS			
Send urgent mail immediately	●	○	○
Limit message transfer size	○	●	●
Inhibit remote mail receipts	○	●	○
Send mail via intermediate post office	●	○	●
Redirect mail to another user temporarily	●	○	○
Compress message during modem transfer	○	○	●
Support Telebit Trailblazer	○	○	●
Support Hayes V Series 9600	○	●	●
Support external delivery agents	●	○	●
Limit message hop count	●	○	○
ADMINISTRATIVE FUNCTIONS			
User privilege levels	○	●	○
Administrator mail purge	○	●	○
Auto name distribution	●	●	○
Hierarchical naming (User@Group@Site)	●	●	○
Mail groups supported	●	●	●
Extended user names in directory	●	○	●
Realtime mail transfer summary screen	●	●	○
Mail archive function	○	○	○
NAME-SERVICE REPORTING			
List local users	●	●	●
List remote users	●	○	●
List group names and members	●	●	—[a]
List remote post office names and addresses	●	○	○
List remote configuration details	●	○	○
MESSAGE-BASE REPORTING			
Total number of messages in message base	○	●	●
Total number of attachments in message base	○	●	○
Storage in bytes for messages	○	●	●
Storage in bytes for attachments	○	●	○
Number of mail items stored for each user	○	●	●
Number of attachments stored for each user	○	●	○

	CONETIC SYSTEMS	CONSUMERS SOFTWARE	PCC SYSTEMS
Group mail storage by items and bytes	○	○	—[a]
Mail header report for all stored messages	○	○	●
COMMUNICATIONS REPORTING			
Total bytes/items sent to date	●	○	○
Number of items sent to remote post offices	●	●	○
Number of bytes sent to remote post offices	●	○	○
Number of calls made to remote post offices	●	○	○
Number of retries to each remote post office	●	○	○
Number of urgent items sent	●	○	○
List current call table settings	●	○	●
OTHER REPORTING			
Reports by period (month, quarter, etc.)	●	●	○
Reports by year-to-date	●	●	○
Mail-server activity log file	○	●	●
Mail-server error reporting	●	○	○
Cost reporting for long-distance calls	○	●	○
Distribution-list report (groups, members)	●	●	○
Print user account detail (privileges, etc.)	●	●	○
USER FEATURES			
NETBIOS mail notification	○	●	○
Mail priority levels	○	●	○
Mail routing history available to user	●	○	○
User bulletin boards	○	○	●
Private distribution lists	●	○	●
Background mail transfer (remote users)	○	●	○
Mail retrieved by key word	●	○	●
Message expiration date	○	●	○
Cut-and-paste messages	○	●	●
TSR user mail program	○	●	○
Graphics in mail messages	○	○	●
Encrypt attachments	○	●	●

● = Yes   ○ = No   [a] Groups not supported.

None of the three reviewed programs provides all the features required for the ideal wide area mail system. Administrative reporting, an area so important to the success of large configurations, lacks critical capabilities in all three programs. Distribution of strengths and deficiencies are such that a full-featured system can be envisioned only by combining features of all three. Each company promises to provide missing capabilities in the future; for now, careful development of selection criteria is required to match the correct E-mail product with a given site.

Mail takes the answer and forward features one step further than other reviewed products by recording the history of each message routed through more than one user. While reading an incoming mail message, a user can pop up a history window displaying where the message originated and the users the message passed through on its way to the current reader.

Considering that the quality of a mail system's user interface is a somewhat subjective value, anyone interested in E-mail should try Higgins Mail, and other mail packages for that matter, to see if they like it. Because of its origins as a LAN productivity tool, Higgins Mail may be best suited for small to medium wide area mail systems, but large configurations are certainly plausible. Conetic Software should consider keeping its user interface and administrative functionality, but rewriting the underlying database constructs to optimize them for mail.

Although Higgins Mail has some deficiencies, it provides a rich set of features, many of which are unavailable in other E-mail packages.

FLEXIBLE COURIER

Much of The Network Courier's appeal stems from its user interface. Many users are initially attracted by its ease of use and the TSRs that allow background mail transfer and mail hot-key capabilities. The hot-key feature allows a user to enter the mail program without exiting the current application.

While its user services are strong, its wide area functionality is not of the same caliber. No support for indirect routing and multiple hops may prove problematic for complex configurations with a variety of communications links. Still, The Network Courier has powerful and unique features that set it apart from other E-mail products.

The Network Courier runs on LANs supporting DOS 3.1 or later. In addition to being sold through systems houses and dealers, it is distributed under special licensing agreement by AT&T, Tandy, Xerox, and others.

The Network Courier is sold in several modules, purchased separately. The main group of post-office programs is available in three versions: Six-node Starter Kit, $295; Single Network, $695; or Internetwork, $995.

Major components of The Network Courier post-office software are Admin, the administrator's program; Mail, the end-user program; and External Mail, the mail-server program. Full mail-server capabilities are available with Mail Postoffice, priced at $495.

One more add-on product for The Network Courier is the $1,000 PROFS Gateway, which makes The Network Courier post office look like a VM node to a host-based PROFS. With this product, PROFS users can view a listing of the names of The Network Courier users and send mail to them as if they were a remote VM node. One implication of The Network Courier's transparency to a VM network is that its post offices can use VM leased-line backbones to store and forward mail across wide geographic areas.

The Network Courier's Internetwork version supports remote mail users with a unique add-on module called Modem Mail, available for $95. One option of Modem Mail, LISTEN, is an 85KB TSR program that runs as a background process in a remote mail user's PC. LISTEN monitors the COM port for calls from another remote user or post-office mail server. When a call is received, a background transfer of mail takes place. This eliminates the need for the remote user manually to initiate communications with other nodes in the mail system.

Included with The Network Courier's basic post-office software is another TSR program called Mail Monitor that alerts the user to incoming mail. Whereas most other mail-notification programs poll the message base for

new mail, Mail Monitor uses NETBIOS to transmit mail notification directly from the sender's workstation to the receiver's workstation. This results in almost instantaneous notification. If NETBIOS is not loaded, Mail Monitor defaults to a polling method.

Mail Monitor offers a powerful capability, unique among the reviewed products: users can hot-key into the mail program to compose or read mail—without exiting the foreground application. For example, if users find the internal mail editor inadequate, they can compose messages in their word processor, hot-key into the mail program, and send them without exiting the word processor.

The trade-off for this powerful feature is 54KB of workstation RAM. Users who cannot allocate this much memory for a mail notification program can turn to Micro Mail Monitor, which functions with a meager 6KB of workstation RAM. This smaller program notifies the user of mail reception, but does not allow hot-key access; the user must exit the current application when notified of mail in order to execute the mail program. In *PC Tech Journal* tests, Micro Mail Monitor did not perform reliably and gave erroneous notification of mail when none existed.

The administration menu—as with all The Network Courier menus—is driven by Lotus-1-2-3-style command bars. Cursor keys are used to highlight selections, the Enter key selects, and the Esc key backs out. There are selections for adding and modifying user accounts, defining group, managing the message base, managing passwords, and exporting a user list.

The Network Courier possesses a rich set of user privileges. Users can have send-only or receive-only status; they can be proscribed from the urgent-mail and remote services; and they can be prevented from deleting messages. cc:Mail and Higgins Mail cannot limit user privileges in this way.

The Network Courier's message-base management routines are run periodically to maintain file integrity and purge mail messages. A compression routine repacks the message base and reports the amount of space recovered in the process. The delete function purges mail based on several criteria: age; priority level; specific users; read or unread. These criteria allow the mail administrator to selectively purge mail, thereby minimizing the disk storage space occupied by outdated mail. The delete function can be used in conjunction with storage reporting, which describes usage in terms of bytes used by all users, bytes used by mail files, and total bytes used by the system.

Mail system overload can also be controlled in part by the "useful life" function. Any outgoing mail item can be assigned a useful life, measured in hours from 1 to 999. The system deletes this item when its useful life has elapsed—whether or not it has been read. This feature is especially useful for messages informing users of staff meetings, office parties, and other time-sensitive events. The Network Courier does not support bulletin boards or private message folders for mail management as does cc:Mail.

Mail Monitor users can hot-key into the mail program to compose or read mail without exiting the foreground application.

Some mail management can be achieved by storing messages in a user-created set of DOS subdirectories.

An export feature allows the mail administrator to update name services on other post offices. This process is normally activated after local modification to user and group accounts. Once requested by the administrator, this process is automatic; either all post offices or specific ones can be updated.

Mail-server software manages inter-post-office and remote-user communications in The Network Courier system. As with the other E-mail products, The Network Courier's mail server can transfer mail to external post offices through the phone lines or with virtual drives on an internetwork.

The Network Courier does not support indirect addressing, as do Higgins Mail and cc:Mail. This means that centralized configurations where hub post offices manage inter-post-office routing are not possible. Peer-to-peer configurations are possible with The Network Courier. These can be quite extensive but may have management problems if they get too large.

The Network Courier mail server is a dedicated PC with a modem. Additional mail servers can be installed to increase the throughput. No additional products are required to add mail servers to a post office.

Although the Network Courier documentation indicates that it supports the Telebit TrailBlazer, an investigation proved differently. The TrailBlazer is fully supported only if the communications software can drive the serial port to 19.2 kilobits per second (Kbps). There are some very old PCs that will not support 19.2, but most everything manufactured since 1984 will, including 80286s and 80386s. The Network Courier's mail server will run the serial port only at speeds of up to 9,600 bps, effectively throttling the TrailBlazer at half its attainable throughput. The Telebit Corporation maintains that its modem can support *averaged* speeds in excess of 14 Kbps for mail servers and other high-speed communications applications.

The Network Courier keeps statistics on its mail-server activity. The number of messages sent and received and a sequential record of each call the mail-server places are displayed on the mail-server screen. Although these records scroll off the screen as new calls are made, they are also stored in a log file that preserves a record of all calls placed to local or remote post offices, the name of the post office, the time the call was placed, and modem initialization sequences. This is a helpful record for the mail administrator in managing the routing configuration.

Selections on the administration menu give access to the routing settings, including number of call retries, call retry delay, and maximum message size (in bytes). The default for this last parameter is 0, which means mail item size is unrestricted.

The Network Courier presents a full set of options for specifying the call patterns for remote post offices. The administrator can enter specific times of day or time periods during which calls can be placed from a given post office. These restrictions can be overridden by requesting priority handling for a message. Priority messages are transmitted to remote post offices immediately. Another mail-server parameter determines whether post offices can send and receive mail or just send it during its outbound calls.

For remote post offices that make connections with modems, The Network Courier can prevent receipts for registered mail from coming back through the communications link. The mail administrator may determine that these receipt messages are too expensive to transfer over long-distance connections. This is a unique feature of The Network Courier.

Another unique—and particularly valuable—feature of The Network Courier is its ability to perform cost accounting for its long-distance calls. Toll charges are entered in cents-per-minute for different periods of the day; the software records and reports on these charges for each remote post office.

The editing functions of The Network Courier cannot compare with a commercial word processor, but they are satisfactory for most mail composition. The full-screen editor functions include automatic text reformatting; cutting and pasting DOS files into messages; word wrapping; and importing a DOS file. The Network Courier has no graphics-creation capabilities.

A file management feature similar to Norton Commander or Executive Systems' XTree efficiently handles attachments. When a file is to be attached to a message, the user pops up a window that lists the files in the default DOS directory. This window, called a *tree-walker*, permits the user to traverse the DOS directory structures with the cursor-control keys. The files in any directory on the drive can be viewed with a few quick keystrokes. A press of the spacebar selects highlighted files for attachment.

Once a message is composed and attachments selected, it can be addressed to a user or group of users. A message-priority system conveys to the recipient the urgency of a mail item. Priority levels range from 1 to 5, with 1 meaning low priority and 5 meaning urgent; the default is no priority. A priority 5 message to a user on another post office forces the mail server to contact the destination post office immediately. Notification of the receipt of a priority message is transmitted rapidly with NETBIOS.

The Network Courier uses a three-part name-directory scheme, similar to that of Higgins Mail. The hierarchical subdivision of name directories normally requires users to make a name-directory change when addressing mail to recipients outside their group. With The Network Courier, external mail recipients who regularly receive mail can be assigned an alias that eliminates the need for specifying their full logical address. This saves the effort of moving around in the subdivided name directories to address mail to remote users.

RAPID TRANSIT WITH CC:MAIL

If mail-transfer speeds were the only criteria by which to judge these E-mail products, cc:Mail would come out ahead. Tests show it to be substantially faster than the others. In other respects, however, cc:Mail has its own mixture of strengths and weaknesses.

This local and wide area mail product developed by PCC Systems is designed to run on any PC LAN that supports DOS 3.1 or later. Like many third-party mail products, cc:Mail is packaged in a number of modules that are purchased separately. The essential post-office programs comprise the cc:Mail LAN Package, available in 25-user or unlimited-user versions for $595 and $1,190, respectively. The LAN Package includes an administrator program, end-user program, and a communications utility called DIALIN.

DIALIN runs on a network workstation with a modem, allowing remote users or remote post offices to call in and transfer mail messages. It does not support outgoing calls, or update of local post offices on an internetwork, and thus cannot be considered a true mail-server program.

To achieve full wide area functionality, a separate program called cc:Mail Gateway is available for $1,295. This mail-server program runs on a dedicated network PC and provides both in-bound and out-bound mail-server communications. cc:Mail supports multiple mail servers per LAN; additional copies of Gateway are not required for mail servers on the same post office.

Another mail program sold separately by PCC Systems is cc:Mail Remote. This $295 product must be purchased for each remote mail user; it runs on a stand-alone PC with modem.

A PROFS gateway, cc:Mail PROFSlink, allows cc:Mail users to exchange messages with PROFS users. The messages can have spreadsheets, documents, and other DOS files attached. The current version of this product allows the entry of PROFS user names in cc:Mail name directories, but not vice versa. cc:Mail users are represented by one account name on the PROFS system. All mail sent from PROFS users to this account is routed through the PROFS gateway and deposited in a single mailbox of the cc:Mail post office. Messages for individual cc:Mail users must have the intended recipient's name embedded in the text. These messages are distributed either by giving all users access to a single cc:Mail mailbox, which is problematic, or by manually readdressing the mail messages based on the embedded name.

The Network Courier PROFS gateway is more developed than the cc:Mail product because of its support for mul-

tiple user accounts on either side of the gateway. PCC Systems may add this functionality in the future.

The PROFSlink program is purchased separately for $995 and can run with the mail-server software on a single dedicated PC or on a separate PC. Requirements on the PROFS side are IBM PROFS version 2.2 with REXX interpreter and EDIT; and one PROFS/VM user ID with "G" privileges.

Message-base data files, not included on the diskettes shipped with cc:Mail, are created the first time the administrator program is run. The cc:Mail data management routines are particularly well designed, producing a compact and efficient message base even when many messages are stored. For this reason cc:Mail is well suited to post offices with a large number of users and heavy mail traffic. Installed cc:Mail systems can reliably support message bases of more than 100MB and maintain good response times.

Administrator utilities are provided with the cc:Mail LAN Package for periodic maintenance of the message base. The CHKSTAT utility verifies message-base integrity and adjusts counters and pointers where necessary. A detailed list of users and their message count is output during this process. Summary reports showing total numbers of messages per user are also generated by the CHKSTAT utility.

As with Higgins Mail, the cc:Mail administrator cannot selectively purge mail items from the message base. Another utility, RECLAIM, can repack the message base to recover spaces in the file where gaps left by deleted messages are not completely filled by new messages. Both CHKSTAT and RECLAIM should be run when the mail servers are down and the mail-user program is not being used.

PCC Systems offers a utility for end users to exchange ASCII files with its message base and name directories. This $995 Import/Export utility consists of two conversion programs executed from the DOS command line. With the Export program, messages already created by users and queued for sending may be converted from their encrypted cc:Mail format into ASCII format. Once converted, ASCII messages can be transferred by an external delivery agent or accessed by other end-user programs. The Import program entails the reverse operation, converting an ASCII message file from an outside source into cc:Mail's internal format.

The Import/Export utility can also convert name directories for distribution to other cc:Mail post offices. This is important because, unlike the other products reviewed, cc:Mail does not support automated name-service distribution. For this operation, the Export program outputs an ASCII table of mail-user names and addresses that are then distributed (as a mail attachment) to other post offices. When this table is received by a post office, the local administrator uses the Import function to replace the existing name directories with the entries from the imported table. PCC Systems would do well to add auto-name distribution to cc:Mail in the future.

cc:Mail administrative utilities have convenient pop-up menus. In addition to its menu selections, the initial administrator screen has a status area that indicates how many mailboxes have been created, number of messages in the message base, number of bytes of message-base storage, and number of bytes of total message-base storage. All attachment files are encrypted and stored in the message base along with mail messages.

Routing and communications functions are selected from the administrator's menu system. Here, routing tables containing entries for other post offices are built and modified. Post office addresses in routing tables specify servers, drives, and DOS directory paths for local post offices and phone numbers for remote post offices.

Although it does not support hierarchical naming, cc:Mail is a true store-and-forward delivery system supporting indirect routing, peer-to-peer, central-branch office, or hybrid configurations. The parameters regulating inter-post-office connections are minimum messages for a call; maximum message size; time to call; auto-call interval; number of call retries; retry intervals; and type of connection.

The routing tables can contain multiple time-to-call parameters for a single post office. This makes it possible to customize a unique set of connection times for every remote post office. The type-of-connection parameter specifies whether a remote exchange sends and receives mail, or only sends mail. This is useful for peer-to-peer configurations in which each post office is responsible for its call charges. The maximum message-size parameter sets a ceiling on the number of bytes in a message.

Among the most noteworthy features of cc:Mail is its ability to create bulletin boards and mailing lists. The mail administrator assigns topics to the bulletin boards, which are made available for user messages. cc:Mail supports up to 100 bulletin boards with 500 messages each. Local users can post and read bulletin-board messages, but remote users can only post them—this is intentional. Only the administrator can delete bulletin boards and messages when they become obsolete.

The mail administrator can also create as many as 100 public mailing lists, each containing 40 names of local or remote users. Mail users can address mail to groups of users with public mailing lists, but cannot modify them. Additionally, users can create and modify private mailing lists that are not accessed by other users.

The cc:Mail user program is a full-featured application with many advanced features. The full-screen editor is function-key driven and supports word-processing block operations, insert and delete, search and replace, word wrap, tabs, and margins. The editor can read or write ASCII files and import existing messages during the message composition process. A Snapshot utility captures a screen image and stores it in a file. The snapshot file can be retrieved and inserted in a message or viewed while in the mail program. A drawing package included with cc:Mail generates complex graphics images, bar charts, and free-hand sketches for enhancing message content.

Despite its many advanced features, cc:Mail's user program lacks an especially important function: automatic paragraph reformatting. Without it, long messages are difficult to compose. Edited text that does not fill out the margins must be blocked and manually reformatted. The alternative to composing extensive messages with the cc:Mail editor is to write a short cover message and attach a document file created with your favorite word processor.

A cc:Mail item can have up to 20 attachments, including messages created with the full-screen editor; DOS files; snapshots; graphics images from the drawing program; or previously stored messages. As attachments are added, a status line informs the user of the total bytes in the message.

After a message is created, cc:Mail presents the user with an extensive selection of transmittal options: messages can be sent to another user, group of users, public mailing list, private mailing lists, bulletin board, or post office. For important messages, a receipt-request feature instructs the mail system to notify the sender when the recipient has opened—not just received—the message.

Upon receipt, mail is itemized in a display that lists most recently received mail first. After mail has been read, it can be forwarded to another user with comments; stored; deleted; or placed in a private message "folder" created by users to store and organize their mail. Received messages can be moved from the general storage area into folders on specific areas of importance. A folder can store messages both sent and received by the user. Messages can be moved between folders, viewed, and deleted. cc:Mail can search all of a user's undeleted messages for a key word or phrase, a date range, or the sender's name.

The cc:Mail status window reminds the user how many messages are active, stored on bulletin boards, and in folders. The user should delete unnecessary messages to conserve LAN disk space. If users do not delete the messages they have read, the administrator's reports will reflect that.

A cc:Mail program called NOTIFY alerts a user to waiting mail with a pop-up window superimposed on the user's foreground application. NOTIFY is a TSR module loaded into 30KB of workstation memory during the network log-in process. Interestingly, if NOTIFY is resident when a message generated by Novell NetWare's SEND command is received, the workstation will hang. This behavior also occurs when other applications interact with Novell's SEND command.

As with all TSRs, conflicts must be resolved on a case-by-case basis. When in operation, NOTIFY scans the message base for new mail addressed to the user, rather than responding to a direct NETBIOS alert from the sender's station, as with The Network Courier; NOTIFY's method may cause a short delay between receipt and notification.

Of the three packages reviewed, cc:Mail comes closest to a full-function, wide area mail system, although it lacks some of the desirable administrative functions found in Higgins Mail and The Network Courier. Regardless, the primary concerns for LAN-based E-Mail systems are dependability, security, and performance. cc:Mail meets these criteria to a high degree, providing a mature and stable E-mail environment.

NOBODY DOES IT BETTER

Successful local and wide area mail implementations are possible with all three of the products reviewed here. They all support dedicated mail servers and multiple post offices—remote and local. No clear-cut winner emerges in terms of dominance of end user, administrative, and communications services. The best product would be a hybrid with the strengths of all three.

None of the three packages has completely adequate administrative reporting. The lack of an administrator mail purge function in Higgins Mail and cc:Mail is one of the most glaring omissions in these packages. The absence of call-retry reporting in The Network Courier and cc:Mail is also unfortunate. Users with security needs must carefully consider the fact that Higgins Mail does not encrypt attachments. None of these systems provides a backup connection route to remote post offices in the event of a primary connection failure.

Ideally, a wide area E-mail system should generate a single report detailing traffic levels, number of retries, and

XDB: The Top Rated SQL DBMS

Listen to what the experts are saying about XDB, the speedy SQL database management system that leaves the others in the dust:

"Our scoring system ranked XDB number one for its consistently speedy performance...XDB earned the highest rating overall by outperforming the other tested programs in ease of learning, ease of use, and performance."
Software Digest, Sept. 1987

"...XDB is the fastest database manager ever tested by PC Tech Journal."
PC Tech Journal, April 1988

"(XDB's) superiority over many DBMSs in various comparative benchmarks demonstrates the quality of its optimization..."
Data Based Advisor, Feb. 1988

But speed is only one reason why XDB has emerged as the top rated SQL DBMS. XDB offers:

✓ **Menu-Driven Tools:** We ease application development by giving you a complete set of tools. Use XDB's SQL, Report Writer, Forms Manager, and 4GL Procedure Language to create application components. Then simply integrate them into a complete application with XDB's Menu Generator.

✓ **Sophistication:** You can build even more sophisticated applications using C, PASCAL, and COBOL embedded with SQL. And your single-user applications can be ported to multi-user systems without change.

✓ **Compact:** You would think that with all these features and power that XDB would require loads of memory. Surprise! XDB's SQL processor occupies only 220K. It fits in 640K even with all the application tools.

XDB applications run unchanged on PCs, LANs, and UNIX today. Tomorrow they will run on OS/2 — or even mainframes. Which means your investment today in XDB is protected as you grow.

So what are you waiting for? Get your hands on XDB and see what life is like at the top of the SQL DBMS mountain.

Call (301) 779-5486 today,

Or send the order form below.

XDB's 30-Day Money-back Guarantee.
Try XDB-SQL or XDB-FORMS for 30 days. If you're not totally satisfied, we'll refund your money cheerfully.

Please rush me the following:

___ XDB-SQL: DBMS, Report Writer and tools	$495
___ XDB-FORMS: Forms Manager	$295
___ XDB-C, XDB-PASCAL, XDB-COBOL (each)	$395
___ XDB-LEARN: limited XDB-SQL and XDB-FORMS	$ 39
Subtotal:	______
Md. residents add 5% sales tax:	______
Shipping and handling:	$ 10
Total:	______

☐ Check ☐ Money Order ☐ Visa ☐ MasterCard

Card No: ______________ Expires ______

Name: ______________
Title: ______________
Company ______________
Phone ______________
Address: ______________
City, State, Zip ______________

xdb
XDB Systems, Inc.
7309 Baltimore Ave., Suite 220
College Park, MD 20740

 CIRCLE NO. 136 ON READER SERVICE CARD

calling patterns for all post offices. This type of a report serves as a model of actual communications activity on the mail system as a whole and is necessary for accurate load balancing and performance tuning of large systems. This is similar to functions in IBM's NetView for large SNA networks. None of the third-party E-mail products has this capability.

Both Higgins Mail and cc:Mail support the indirect addressing required for large centralized configurations. The Network Courier supports peer-to-peer configurations, but not indirect addressing. Because many large E-mail configurations require intermediate nodes, lack of indirect routing may eliminate The Network Courier for use in corporate systems that route mail through central hubs.

In *PC Tech Journal* tests, mail-transmission speeds to remote post offices varied substantially. Average transmission times for a one-page message with two-page attachment file were 1 minute 33 seconds for cc:Mail; 2 minutes 31 seconds for The Network Courier; and 2 minutes 39 seconds for Higgins Mail. cc:Mail performs extremely well during mail transfer because of its file compression techniques and therefore saves costs in toll charges. cc:Mail is the only one of the three that supports the Telebit Trailblazer at speeds of more than 9,600 bps.

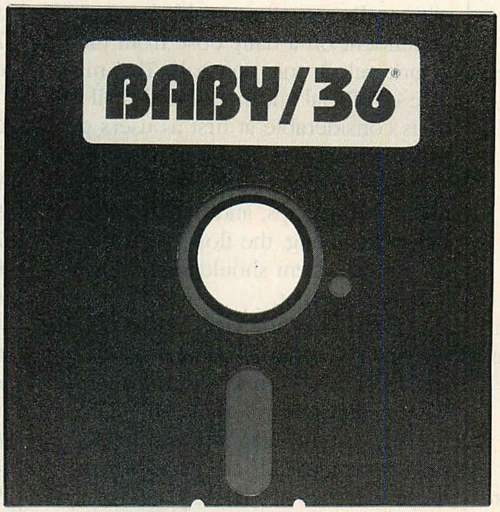
All three companies have tried to cram as many features as possible into their end-user programs. Unfortunately for the end user, these efforts are not completely successful. The line-draw and related graphics functions in the cc:Mail editor are wonderful, but without automatic paragraph reformatting, users will tend to avoid it for extended messages. The next release of cc:Mail is promised to support automatic paragraph reformatting and many mail-editing enhancements.

The key-word search feature of the Higgins package might be useful if not for the restrictive two-page limit on mail messages. The Network Courier arguably provides a good, general-purpose user environment, but its nested Lotus-like command lines can be frustrating for hurried users engaged in rapid-fire memo salvos. There is room for improvement here too.

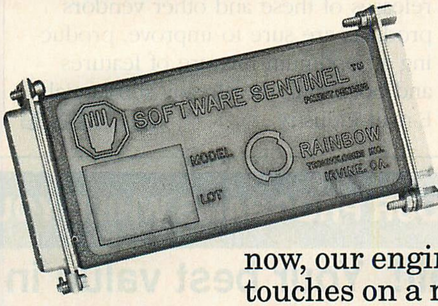
THE MAIL MUST GO THROUGH

Though widely useful, E-mail is not the perfect software solution for every distributed application. Distributed databases, for example, require more than file and message transfer. These data-

This Software Won't Run.



Without This.



And that's the way California Software Products, Inc. wants it. And a host of other big and not-so-big software developers who use our Software Sentinel. To make sure their successful software is protected. So it stays successful.

We've become used to being seen with top-notch software. A lot of the reason is how our engineers designed the Software Sentinel family. For the developer and the user. To make it the most technologically foolproof yet easiest to use software protection key you or CSPI can get.

For users of BABY/36 and the thousands of other programs we're protecting, it's a cinch to get up and running. They simply plug the Software Sentinel into the PC's parallel printer port. That's it. How much easier can it get? Users can even make unlimited backup copies. And run them wherever and whenever they need to—as long as they have the Software Sentinel key.

As long as we're talking success, there's something else you should know. Right now, our engineers are putting the finishing touches on a new microprocessor that we'll use in a whole new generation of software and data protection products. With the same high-performance and high-reliability of our past successes but with many new features. Which no doubt is going to make life even easier for you and CSPI.

The Software Sentinel. Making sure software developers stay successful with the successful software they've developed.

For more information on the Software Sentinel family, contact Rainbow Technologies, 18011-A Mitchell South, Irvine, California 92714; or call (714) 261-0228.

Software Sentinel Features:

- *Runs under DOS and Xenix*
- *Uses algorithm technique, never a fixed response*
- *Minimal implementation*
- *Higher level language interfaces included*
- *Transparent operation*

RAINBOW TECHNOLOGIES

18011-A Mitchell South, Irvine, CA 92714 • (714) 261-0228 • TELEX: 386078 • FAX: (714) 261-0260

CIRCLE NO. 157 ON READER SERVICE CARD

bases often use a specialized front-end program that is a major element of the software's functionality. It is usually not adequate to send a message to a database; the user must run application-specific code in the workstation.

Increasingly, wide area network implementations are deploying remote bridges to link workstations to databases on remote servers. The effect of a remote bridge is to make the local and remote LANs appear as a single LAN to the workstations and the applications. Although this approach is much more expensive than store-and-forward message technology, remote bridging may be a better solution than E-mail for realtime applications such as distributed financial systems.

The export/import facilities supplied with Higgins and cc:Mail have been used in conjunction with user-developed code to provide file-transfer operations for network applications. Distributed order entry with an E-mail front end is a feasible example of this. Generally, however, LAN E-mail packages are best at providing document and file-transfer services for LANs and PC users distributed across large geographic areas. They do not have well-developed APIs for program-to-program communications.

LAN-based E-mail will change the patterns of communication within an organization—mostly for the better. In the early stages of implementation, E-mail systems can be very popular with end users who quickly become dependent on a daily dose from their favorite electronic pen pals. This mixture of official and nonofficial mail traffic is considerable at first as users get to know each other in a new way—one that crosses boundaries between titles, workgroups, and departments. Before too long, the flow of messages on a new system should settle down to a healthy torrent.

Because of the vital nature of interpersonal communications—subjective or not—E-mail systems must have a high level of availability. Consequently, careful planning and implementation methods are necessary. If these methods are inadequate, the system will degrade—and as everyone knows, "the mail must go through."

Higgins Mail, The Network Courier, and cc:Mail would all be justified in proclaiming this old adage. Future releases of these and other vendors' products are sure to improve, producing an optimum mixture of features and bolstering the already substantial E-mail industry.

Conetic Systems Inc.
1470 Doolittle Drive
San Leandro, CA 94577
415/430-8875
Higgins Mail, $495; Higgins Exchange, $1,595; Higgins Remote, $395; Higgins To:3+, $495
CIRCLE 345 ON READER SERVICE CARD

Consumers Software Inc.
73 Water Street, Suite 603
Vancouver B.C., Canada V6B 1A1
800/663-8935; 604/688-4548
The Network Courier: Six-Node Starter Kit, $295; Single Network, $695; Internetwork, $995 per file server; Modem Mail Postoffice, $495; PROFS Gateway, $1,000
CIRCLE 346 ON READER SERVICE CARD

PCC Systems Inc.
480 California Avenue
Suite 201
Palo Alto, CA 94306
415/321-0436
cc:Mail LAN Package, $595 (25 user), $1,190 (unlimited user); cc:Mail Gateway, $1,295; cc:Mail Remote, $295; cc:Mail PROFSlink, $995
CIRCLE 347 ON READER SERVICE CARD

Steven S. King is a technical editor with PC Tech Journal, *specializing in LANs.*

KEEP UP WITH THE OS/'s

Megabytes of Memory and 32-Bit Performance for DOS.

If you thought the only way to protected mode was the big move to OS/2... We have good news! You can gain the benefits of protected mode the easy way with OS/286™ and OS/386™. These tools for C, Fortran, Pascal and assembly language programmers permit rapid conversion of existing DOS applications from "real" 8086 mode to "protected" 286 and 386 mode. They don't replace or modify DOS, but extend it to protected mode. This way you get multi-megabytes of directly addressable memory (16Mb-286, 4Gb-386) with the compiler, TSRs, device drivers, graphic routines, etc. you use today.

OS/286 and OS/386 are the only DOS extenders that span both the 286 and 386 processors. They run on the widest array of AT and 386 machines, with 32-bit capability *today* on 386s that yields twice the performance of 16-bit mode.

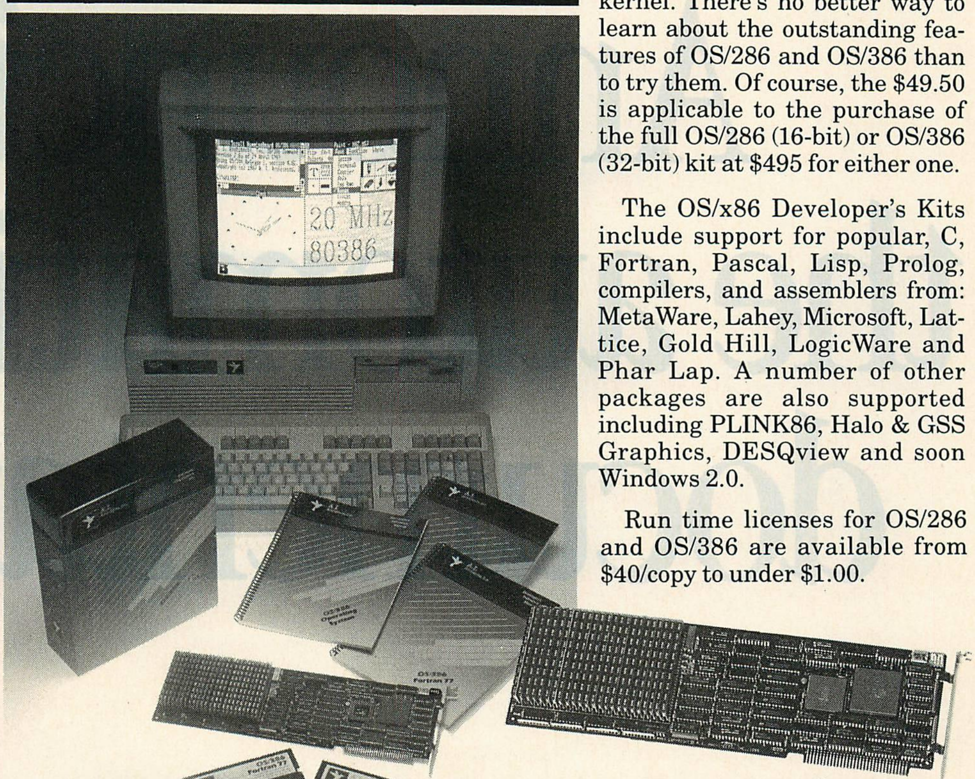
OS/286 and OS/386 are the preferred solutions for developers of high performance memory-intensive applications, including CADKEY, CASE, and Gold Hill, and premier language developers Lahey and MetaWare.

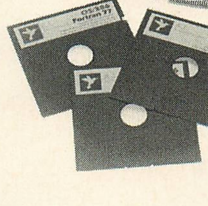
Our optional TOUCHDOWN™ BIOS supplement provides fast and reliable protected mode operation on *any* 286 system, even those with problems resetting the 286. (Ever notice how few existing machines Operating System/2 runs on?) TOUCHDOWN is not required for most major brand AT clones, but for the older machines it is a lifesaver!

If your applications are running out of memory or need more speed, enhance them now without abandoning your investment in DOS.

Coming Soon
WINDOWS 2.0
and DESQview 2.0 API support

Give your protected mode DOS a powerful presentation layer like OS/2, SAA or DESQview.





Special $49.50 Evaluation Offer

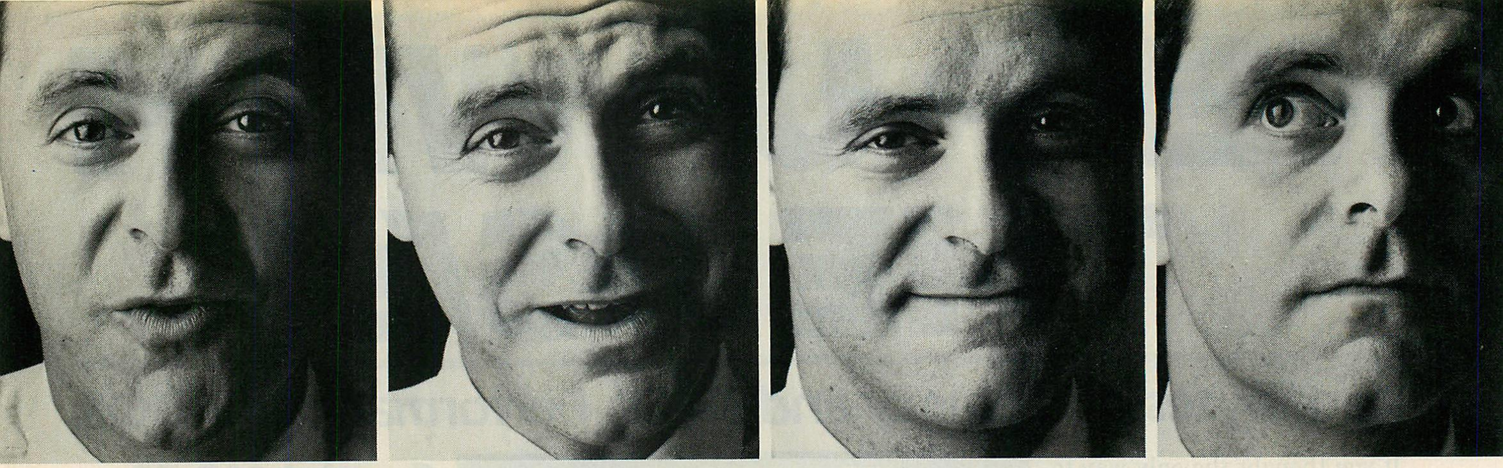
Check out for yourself the benefits of protected mode. Our $49.50 "sampler special" is a complete OS/286 Developer's Kit, but with a time limited, non-distributable kernel. There's no better way to learn about the outstanding features of OS/286 and OS/386 than to try them. Of course, the $49.50 is applicable to the purchase of the full OS/286 (16-bit) or OS/386 (32-bit) kit at $495 for either one.

The OS/x86 Developer's Kits include support for popular, C, Fortran, Pascal, Lisp, Prolog, compilers, and assemblers from: MetaWare, Lahey, Microsoft, Lattice, Gold Hill, LogicWare and Phar Lap. A number of other packages are also supported including PLINK86, Halo & GSS Graphics, DESQview and soon Windows 2.0.

Run time licenses for OS/286 and OS/386 are available from $40/copy to under $1.00.

The HummingBoard® turns any XT, AT or 386 into the fastest, most capable 386 system available. The dual processor architecture provides up to 900K each for multiple real mode applications (Lotus, etc.) which can co-reside with protected mode applications on the HummingBoard's 20MHz 386, while debuggers, editors, networks run concurrently on the base processor. 1-24MB memory, 8,870 Dhrystones. A MUST TRY!

A.I. Architects, Inc.
One Kendall Square, Cambridge, MA 02139
TEL (617) 577-8052 FAX (617) 577-9774
Credit card orders only – 24 hours. 617-577-1305

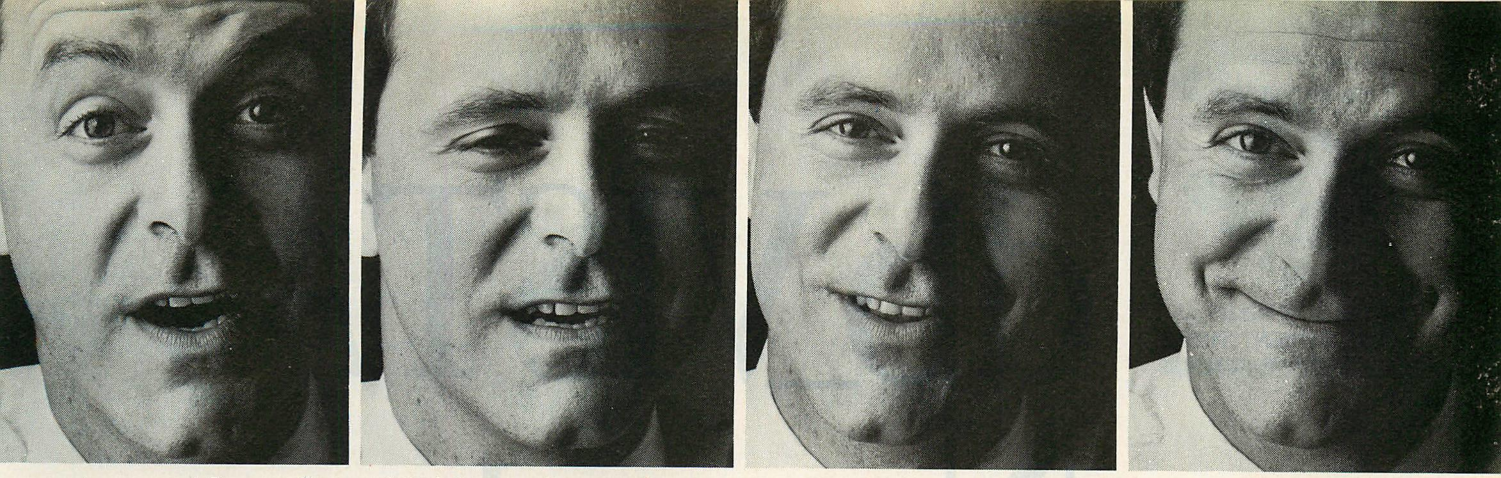


Before: *"Who needs documentation? I know what I did. I think."*

Announcing the automation of documentation.

Canada (416) 231-1270 • United Kingdom 01-554-0582 • Italy 2-439-0771 • Benelux 73-414855 (Holland) • Scandinavia 71-46166 (Norway)

Switzerland 42-415656 • West Germany/Austria 2174-5015 • Australia 02 327 8162 • Brazil 11-881-0600 • France 1-4361-5018 • Portugal 2185098



After: *"I couldn't have done it better myself if I did it myself, if I say so myself."*

Only from DataEase.

When personal computing was still personal, documentation wasn't a corporate problem.

But today, Information Centers are responsible for hundreds of different PC applications written by hundreds of different end users. As well as for developing new applications.

And with the pressure of new development, even professionals don't always have the time to document applications properly.

It's a disaster waiting to happen.

The solution is a tool that lets you create professional applications and documentation without the drudgery. And do a lot more work in a lot less time.

DataEase Developer.™ The only system that automatically documents the applications you develop.

DataEase Developer automatically creates documentation that can satisfy the most stringent MIS requirements. Yet people who don't even know what documentation is can create it quickly and easily.

Total system documentation, including all forms, reports, menus, relationships and definitions, is created automatically. And user manuals can be created just as easily. DataEase Developer outputs to Ventura Publisher® and any word processing program that reads ASCII file formats.

DataEase Developer provides everything needed to build complete professional applications with evaluation copies, self-running demonstration disks, encrypted applications and serialized versions to protect applications from unauthorized use.

And developing applications is easy. Because DataEase Developer is built around DataEase,® the database manager that's as powerful as it is easy to use. And The DataEase Family of Products™ provides a total solution for end user computing. From connectivity to database to graphics.

DataEase Developer. Because everybody needs documentation but nobody wants to do it.

Call 1-800-334-EASE

(in Connecticut, 374-8000) for more about DataEase Developer.

DataEase INTERNATIONAL

Tools for the applications generation.

db_VISTA's Network Approach

ANDREW TOPPER

The bigger the database the better for db_VISTA. Its mainframe-like network data model eliminates the duplication of data inherent in relational systems, and its support of C turns db_VISTA into a portable, powerful development environment for the PC.

In the world of PC data management systems, relational products, or those that claim to be, dominate the marketplace. But some developers have found working with the relational model difficult, requiring duplication of data and extra effort in implementing complex applications.

The relational model is not the only solution. Taking its cue from the mainframe and minicomputer marketplace, db_VISTA from Raima Corporation is based on the network data model used by mainframe data managers such as Cullinet's IDMS and Cincom's TOTAL. It gives developers a powerful environment, and, because it is written in C, it is transportable.

A data manager based on the network data model contains many of the tools familiar to mainframe developers: data dictionaries, built-in transaction processing, standard programming language interfaces, and automatic database recovery. With these tools in hand, mainframe developers can more easily manage complex problems such as project control and multiuser/multitasking environments. As PC developers begin to face these concerns, the network model becomes more alluring.

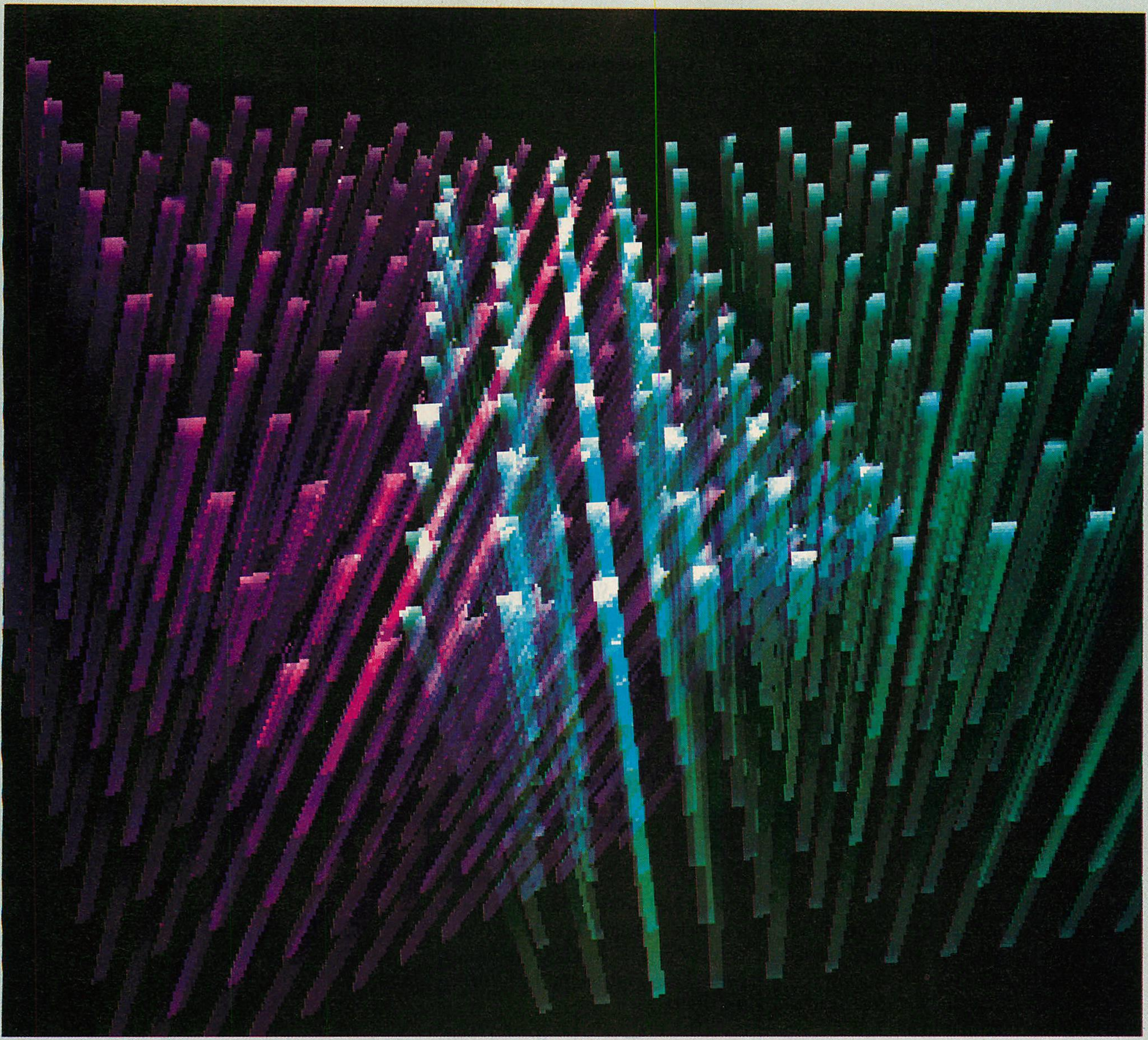
Initially proposed by the Database Task Group (DBTG) of the Conference on Data Systems Languages (CODASYL) in 1971, the network model gained widespread acceptance throughout the 1970s in the mainframe world. The db_VISTA package is one of its first forays into the PC's domain.

The network data model, is based on logical records referred to as *record types*, fields referred to as *data items* or *elements*, and the physical records in a database file referred to as *record occurrences*. The basis for relating record types in the network model is the *set*.

Record occurrences are often physical records in a file with the sets represented as links between the records. A set is a logical definition of the relationship that exists between records, which may or may not be in the same file. The network model supports one-to-one, one-to-many, and many-to-many relationships between records.

In the network model, records within the database can be accessed through a common field known as a *key*, as well as through sets. Program views in the network model are known as *subschemas*, and the overall database is defined in a *schema*. The basis of a schema is database definition language commands that are usually independent of any programming language. (See the sidebar, "db_VISTA's Network Data Model.")

The network data model's power lies in its ability to represent complex



COMPUTER GRAPHIC • DOV JACOBSON

relationships between data while limiting the duplication of critical information. In the relational database model, records (which can be thought of as rows in a data table) are related through keys. These key values must be duplicated when tables are *joined*, and the duplication of this data propagates as more tables are created.

The problem inherent in this method is that when duplicated data are to be modified, each table and row containing the key data must be modified. The overhead for such a search presents serious data-integrity problems if the data are not synchronized. The network data model eliminates the overhead of duplicating key data by providing a means of directly relating records. This implies that when key data change, only a single occurrence of the data must be modified. As databases become larger, the benefits of the network data model compared with the relational model become increasingly obvious.

Because db_VISTA uses the C programming language, it allows complex applications to be developed using the data model that many mainframe application developers use. By providing this support through C, db_VISTA also supports a host of operating environments, including UNIX, XENIX, and DEC's VMS. This places db_VISTA in the same category as Faircom's C-tree/R-tree, as well as SoftCraft's Btrieve and similar products that provide software developers with data management support through a structured programming language (see "A Data Manager with Language Flexibility," Burks A. Smith, October 1987, p. 104, for a review of Softcraft's Btrieve).

The basis for these products is linkable object code that can be called from the program using the underlying structure of the data management software with its inherent strengths and ease of use. The result gives developers the tools for complex data processing.

Raima's db_VISTA 3.0 comes in single- and multiuser packages, and the source code for the db_VISTA system is available. The multiuser version supports NETBIOS-compatible LAN products. The multiuser package also provides a lock manager and file- and record-locking. The lock manager is a stand-alone program that coordinates access to records and files for all programs and is thus handled automatically by db_VISTA. The lock manager is a background resident program that runs on the file server in a network configuration. In a UNIX environment, the lock manager is executed as a background process and is usually initiated through the start-up procedure. In

addition, the multiuser version supplies time stamping, transaction processing, logging, and recovery features, which are also available in the single-user system. These tools ensure data integrity regardless of the type of system.

TRANSACTIONAL ANALYSIS

Raima's db_VISTA supports multiuser systems through the use of transactions, record and file locks, and database recovery. A transaction in db_VISTA is based on a series of database updates that must be applied together to maintain the integrity of the database.

For example, consider the task of updating the quantities on a series of line items on a purchase order in an inventory system. The basis for the transaction might be that unless all the line items in the order are processed, the purchase order cannot be processed. One method would be to check each inventory item quantity for all the line items on the order before updating any of them and lock each one, but a better way exists.

With db_VISTA's transaction concept, the transaction begins, and if each line item can be updated in sequence, the transaction ends. If, however, one item has insufficient quantity to be processed, the transaction is aborted (undoing updates performed to that point), an error message is issued, and processing begins on the next purchase order. The db_VISTA package provides commands for transaction processing and supports automatic rollback of data updates from a log file when a program ends abnormally without issuing a transaction end command.

If a transaction fails, either because of a database error or a program crash, the lock manager recovers the updates from the log file and restores the contents of the modified pages before giving another program access to those pages. The log files are supplemented by a system-wide transaction log file that db_VISTA maintains. The log file provides for recovery when the lock manager itself crashes. When this occurs, the contents of the system log file, called the transaction activity file (TAF), recover database updates when the system is restarted.

An intrinsic part of the transaction concept is users will sometimes try to access the same records and files. db_VISTA prevents such contention by allowing for a time-out interval to be established for each lock request. If the lock manager does not grant a requested lock in the time interval used, the requested transaction is halted and the manager returns a status of unavailable to the application. The developer can set the time interval; the default is 10 seconds. This detection and prevention of "deadlock" is critical to multiuser applications.

db_VISTA also provides file and record locks implemented through the lock manager program. While both are supplied, Raima suggests that file-level locks provide a complete multiuser locking mechanism without the need for record-level locks. Although record-level locks are available with db_VISTA, they require all programs to check the status of these locks before updating a record.

The file locks, on the other hand, are maintained and checked by the db_VISTA lock manager. In the case of IBM's PC Network, the lock manager runs on every node in the network. With a NETBIOS-compatible network, the lock manager may not detect a program abort, so another program, CLEARDB, informs the lock manager of the abort.

A major selling point is that db_VISTA supports several operating environments and can be transferred to other systems.

The db_VISTA program provides for automatic recovery through transaction-processing, the lock manager, and log files for each program updating the database. Whenever a program requests an update to a db_VISTA database, the contents of the page containing the modified record are written to a log file unique to the program.

Automatic database recovery has been used in mainframe data managers practically since they first appeared in the late 1970s. The need for automatic recovery is now becoming apparent in LAN and multiuser data managers, although few software products provide it. Without recovery capabilities, the integrity of the database becomes questionable when programs fail.

ESSENTIALS

The db_VISTA program files come on five 360KB diskettes in object code and three 360KB diskettes in source code. The source code is archived using a proprietary compression scheme; the archive programs ARC and DEARC are included on the source code diskette. During testing, installation took less than 20 minutes and was fairly self-explanatory.

One of the major selling points to db_VISTA is that it supports several operating environments and therefore can be transferred to different computer systems. Version 3.0 runs on DOS, UNIX System V, Sun OS, Berkeley 4.2 version of UNIX, XENIX, and VMS. Raima reports that OS/2 support will be available soon. In addition, Univation's Lifenet, Novell's NetWare, IBM's PC Network, 3Com's 3+, and other NETBIOS-compatible LANs are supported.

Raima offers versions of db_VISTA for Lattice, Microsoft, Borland Turbo, Aztec, XENIX, and UNIX C compilers, but the Turbo C version was not available at the time of this review. The review was performed using Microsoft C 5.0, with version 4.0 compatibility selected on installation.

Documentation for db_VISTA includes a *User's Guide*, a *Reference Manual*, and assorted release notes. The *User's Guide* includes an excellent description of the network data model with examples and diagrams that would benefit anyone interested in learning more about it.

The *Reference Manual* describes all the db_VISTA utilities, runtime functions, and the status and error codes. The *User's Guide* presents a sample db_VISTA database including DDL (data description language) commands and C code for screen-entry programs. Technical support for db_VISTA includes telephone support and a forum on Byte Information Exchange (BIX).

INTERFACING WITH C

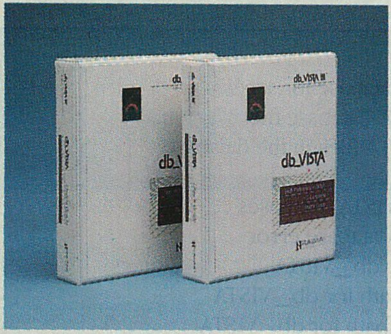
The popularity of C suggests that this product will be of interest to those developers wishing to use an established data manager without creating their own file control routines. Raima's db_VISTA provides an interface to C with compile header files, linkable object libraries, and optional source code for all the db_VISTA components.

When a db_VISTA database is generated through the schema compiler, two files are created: a database dictionary file used by the runtime routines and by some of the db_VISTA utilities, and a database header file containing the record and key structures needed to compile a C program using the database in question. The header file also contains field, record, set, and file constants for the database used in-

db_VISTA OVERVIEW

db_VISTA 3.0
Raima Corporation
3055 112th Avenue NE
Bellevue, WA 98004
206/828-4636
CIRCLE 337 ON READER SERVICE CARD

Product description. db_VISTA provides C-linkable routines that support access to a network model database, utilities, and an SQL-based, ad-hoc query and report-writing facility.
IBM PC environment. db_VISTA works on an IBM PC/XT, PC/AT, or 386-based computer running DOS 3.x. Lattice, Microsoft, IBM, Aztec, and Borland Turbo C compilers are supported. (Turbo C was unavailable for review.)
Other environments. db_VISTA also runs on UNIX System V, Sun OS, XENIX, Berkeley 4.2 version of UNIX, and the VMS operating systems, and is to support OS/2 in the future.
Network support. db_VISTA is available for local area networks that support NETBIOS. These include Univation's Lifenet, Novell's NetWare, IBM's PC Network, and 3Com's 3+.
Copy protection. Not copy protected.
Documentation. db_VISTA comes with two manuals—a *User's Guide* and a *Reference Manual*—in standard-size binders. db_QUERY comes with a *User's Guide* as does db_REVISE.
User interface. db_VISTA provides no set user interface other than the normal DOS command prompt but does include tools for developers and sample database structures and programs.
Help facilities. No help facilities.
File capacities. Maximum record length is limited only by available RAM. Maximum records per file are 16 million and maximum index and data files available are 255. File size is limited only by available disk space; the number of records or sets in a db_VISTA database is not limited.



Data types/capacities. db_VISTA supports all standard C data types, dependent upon the C compiler used. This review used the Microsoft C 4.0 compiler version, which included signed and unsigned integer, signed and unsigned short integer, signed and unsigned long integer, signed and unsigned character, floating point numeric, double precision numeric, and long double precision numeric. In addition, string, pointer, and structure data types are supported.
Data entry. db_VISTA provides no inherent field or record editing capabilities, but the documentation does show how to use db_VISTA records and key files to implement data validation in an application.
Application development facilities. Because this product works with C, the components of the specific C compiler used dictates the development facilities available.
Security. db_VISTA provides no security or data encryption.
Access to system facilities. Access to the operating system is available through standard C calls provided by the compiler.
Reporting. Raima provides an SQL-based product, db_QUERY, at an extra charge, which allows relational access to any db_VISTA database. These db_QUERY routines are C-linkable and can be incorporated into any C program. db_VISTA provides for sorted sets and indexed (keyed) access to records plus a utility for creating key files when additional key fields are required after the database is in production. db_QUERY allows for standard SQL reporting and can use .QRF files, macros, and report formats external to db_QUERY itself. Also provided are PAGE and DATE fields that can be incorporated into db_QUERY reports.
Utilities. db_VISTA comes with utilities to initialize database files, allow interactive access to db_VISTA databases, check the consistency of database files, clear db_VISTA record and file locks, create key files from existing data, check data field alignment, print a database dictionary report, print a dump of key files and data files, and import and export to and from ASCII and dBASE format files.
Data compatibility. db_VISTA supports importing and exporting from/to ASCII and dBASE formats.
Distribution. db_VISTA is available from Raima and distributors.
Price. db_VISTA single user is $595 without source code and $990 with source code; multiuser is $990 without source code and $1,980 with source code; db_QUERY single user is $595 without source code and $990 with source code; multiuser is $990 without source code and $1,980 with it; db_REVISE is $595 without source code and $1,980 with source code.
Support. db_VISTA support is available for 60 days after purchase on BIX (Byte information exchange) and from Raima's technical support. An Extended Applications Development package is available increasing the support and including use of the Raima bulletin board system as well.
—Andrew Topper

ternally by the db_VISTA routines. Once these files have been created by the schema compiler, C programs can be written to access the database through the db_VISTA functions. (See table 1 for a list of db_VISTA commands.)

The db_VISTA commands are coded into the C source and provide access to database resources. The C source must include references to the database header file as well as the db_VISTA header file, VISTA.H. When the program is compiled, the resulting object file is linked with the db_VISTA runtime library VISTAS.LIB to include the functions referenced in the program in the executable file. The contents of the header file depend upon the type of compiler used.

Two concepts critical to programming with a network database are currency and navigation. Understanding how each of these concepts works is essential to using db_VISTA effectively. Currency refers to the identity of the last record occurrence processed for each unique record type in the database. The db_VISTA program maintains currency tables of each record and set the application uses.

In the sample editorial database used for this review, assume an Article record is to be stored in the database. Each Article record is associated with the issue in which the article appeared, identified by the issue volume and number. After storing the Article record, the Issue record with the volume and number for the article has to be

TABLE 1: *db_VISTA Commands*

COMMAND	FUNCTION
GENERAL	
d_open	Open a db_VISTA database for access
d_initfile	Initialize a db_VISTA database file
d_initialize	Initialize all db_VISTA files in the database
d_renfile	Rename a database file
d_close	Close all open db_VISTA databases
dberr	Report a db_VISTA database error
d_dbfpath	Set database path for db_VISTA
d_dbdpath	Set data dictionary path for db_VISTA
d_dblog	Set database log file path for db_VISTA
d_dbuserid	Set user ID for db_VISTA
KEY ACCESS	
d_keyfind	Find the record with the key specified
d_keyfrst	Find the first record occurrence for the key value specified
d_keynext	Find the next record for the key specified
d_keyprev	Find the prior record for the key specified
SET ACCESS	
d_findco	Find the current owner of the set specified
d_findfm	Find the first member in the set specified
d_findnm	Find the next member in the set
d_findpm	Find the prior member in the set
d_findlm	Find the last member in the set specified
RECORD	
d_recread	Read the current record
d_crget	Return the database address of the current record
d_csmget	Return the database address of the current member record
d_csoget	Return the database address of the current owner record
d_crtype	Return the record type of the current record
RECORD MODIFICATION	
d_makenew	Create an empty slot for the record type specified
d_fillnew	Create and fill a new slot for the record type specified
d_keystore	Add the key entry to the key file
d_recwrite	Write the record to the database
d_delete	Delete current record
SET MODIFICATION	
d_connect	Connect the current record to the set
d_discon	Disconnect the current record from the set
d_disdel	Disconnect and delete the current record from the set
LOCK	
d_lock	Lock a group of records or sets
d_rlbset	Set a record lock bit
d_rldclr	Clear a record lock bit
d_setlock	Set a set read-lock
d_setfree	Free read-lock
d_freeall	Free all read-locked files
d_recfree	Free read-locked files specified
d_keylock	Lock files associated with the current key type
TRANSACTION	
d_trbegin	Begin transaction
d_trend	End transaction
d_trabort	Abort transaction
d_recover	Recover failed transaction

When a database is generated through the schema compiler, the compiler creates a database dictionary for use by runtime routines and utilities, and a database header file of record and key structures. Once these files are created, C programs can be written to access the database through the db_VISTA command functions.

located. The Article record can then be connected to the Issue_article set with the correct Issue record as its owner. If not, member records could be associated with the wrong owner records and thus invalidate the database. (For a complete description of the sample application, see "Evaluating Data Managers as Development Tools," Julie Anderson, August 1985, p. 46.)

Navigation refers to accessing database record occurrences through the sets in which they are contained. An example would be to find the issue volume and number that contained an article written by "Bob Smith," again using the editorial database from the sample application.

To retrieve this information, it is necessary to: locate the Author record associated with the key "Bob Smith;" locate the first junction (intersection) record in the Author_junction set; locate the owner Article record through the Article_junction set; and locate correct owner Issue record through the Issue_article set. In this fashion, navigation through the database accesses the records to satisfy the programming requirements. Before creating a program that accesses a db_VISTA database, developers must consider the navigation and currency requirements and create the programs accordingly.

USEFUL UTILITIES

Included with the package are several utilities to maintain databases created with db_VISTA. Two utilities, db_QUERY and db_REVISE, must be purchased separately (see the accompanying sidebar "db_VISTA Overview" for prices).

db_QUERY. Given the recent fervor over Structured Query Language (SQL), it is understandable that Raima provides a pseudo-relational product with db_VISTA called db_QUERY. This product provides a relational view of any db_VISTA database and also supports a subset of SQL.

The existence of this product points to SQL's popularity and the need for data manager vendors to supply some level of support for them to be competitive in this industry (see the December 1987 cover suite "Speaking SQL," specifically "Lingua Franca for Databases," Richard Finkelstein, p. 52). Although, the query-by-example capability in a product such as Borland's Paradox is a much better report writer and more convenient for end users than SQL (see "The Multiuser Perspective," Dave Browning, February 1988, p. 114), some users still demand SQL. For those developers who want to pro-

vide some SQL support, the db_QUERY product from Raima should fit the bill.

At the time of this review, db_QUERY 2.0 was available only in a beta test version. The production version is likely to be functionally the same as that described here.

With db_QUERY, the user can run ad hoc or static reports against any db_VISTA database with the output directed to the screen, a printer, or to an ASCII file (see photo 1). Most SQL commands are supported, excluding those commands that modify the database. Raima states that update capabilities will be provided with the next major release of db_QUERY.

The major SQL retrieval commands supported within db_QUERY include the SELECT command and all of

> ***Groups of fields can be designated with the DEFINE command and can be used within db_QUERY to simplify report content.***

its associated clauses, the mathematical commands COUNT, SUM, MAX, MIN, and AVG, along with the standard report fields PAGE and DATE. db_QUERY has no built-in query optimizer function, and it is up to the developer to establish RELATIONs and PATHs to any records that cannot be located through a key value.

The developer can establish default values for db_QUERY variables in a query initialization file (.QIF) for each database. The commands available in this file include the RELATION command, which designates PATHs to records through existing sets or by linking index fields to fields in other records. With the RELATION command, pseudo-relational views of the db_VISTA database also can be set up.

In addition to this command, the .QIF file can contain field aliases, field type definitions for display, and value/code translation. Also, SET commands establish page length, page width, date format, output format, and db_QUERY prompt. Groups of fields can be designated with the DEFINE command and can be used within db_QUERY to simplify report content.

The DEFINE command can also designate the PATH or RELATION used to access records; this allows the developer to set up common names for groups of records that are normally accessed together. As an example, the .QIF file for the sample editorial database contains a definition called the_works that specifies access to all the articles within an issue and to the authors of those articles.

Any group of SQL commands can be placed into a query routine file (.QRF) and then called from within db_QUERY by supplying the file name at the SQL prompt. This allows the developer to set up static reports that the user can run by supplying db_QUERY with a single name. An added benefit for users would be if db_QUERY could provide a menu of available .QRF files. The .QRF files can also use Report Formats established in files with an extension of .RPT. These formats allow the developer to lay out the content of a report using database fields as well as some of the db_QUERY fields, such as PAGE and DATE.

db_REVISE. One drawback of using the network data model for database applications is that whenever an existing database needs structural modifications, the schema must be changed and the existing data unloaded and reloaded. A separate product, db_REVISE (available only in a beta test version at the time of this review), can redefine relationships within the physical database design to satisfy the ongoing needs of the application.

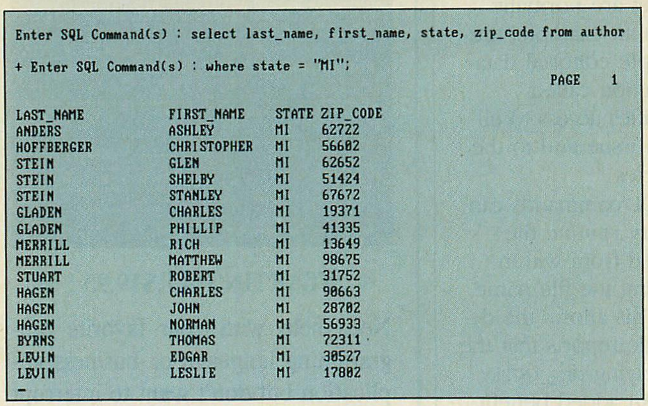
The utility can change field, record, and set names, add record types, remove record types or sets, change the set order, and change the page size for a database file. In some cases, db_REVISE can also improve the performance of a db_VISTA application after it has been implemented.

Some structural changes are outside the capabilities of db_REVISE. For example, db_REVISE cannot create a new set between record types if no set was initially in place.

The db_REVISE utility does not modify a db_VISTA database in place, but rather creates a new copy of the database with new file names and a new database name. In addition, db_REVISE is run after the old schema file is copied with a new database name, new data and key file names, and any other desired modifications by the user; db_REVISE compares the new schema with the old; it then places all differences in a work schema, which is compiled.

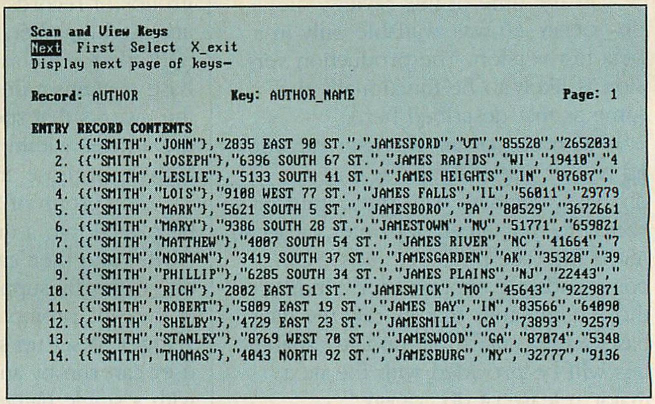
The program then reads through the old database and creates records in

PHOTO 1: *db_QUERY Screen*



In the db_QUERY utility, the user enters SQL commands (or the name of a file that contains SQL commands) in order to generate reports from a db_VISTA database.

PHOTO 2: *IDA Screen*



The interactive database access (IDA) utility provides access to any record or set in a db_VISTA database. Using IDA, records can be added, read, modified, or deleted.

the new database using the work database to match up record addresses and set pointers. Then, using the interactive database access (IDA) utility, the new database can be viewed to determine if all the necessary changes have been made. Another utility, DBCHECK, can check the consistency of the new database. The applications are then changed and recompiled to use the new database structure.

When db_REVISE cannot determine the nature of a change based on the old and new schemas alone, revision definition language (RDL) commands can migrate the old database to the new structure. These RDL commands supply db_REVISE with the information that it needs to determine the differences in structure.

In general, RDL commands are needed when record type names do not match between the old schema and the new schema. If the Author record in the sample editorial database needed to be renamed "Authors," for example, RDL commands would map the Author fields to the Authors fields in db_REVISE. Set names that are to be changed also require RDL commands to implement with db_REVISE.

IDA. Perhaps the most powerful utility delivered with db_VISTA is IDA. This program gives developers access to any record or set in db_VISTA databases without writing any C source code. The menu-driven program supports all db_VISTA commands, but it is designed for the experienced user.

The utility was especially useful in developing the sample application. Through this program, records can be accessed by key or by set, or they can be added, connected, modified, or deleted. In addition, the program supports some simple editing of data fields in records (see photo 2).

For the developer, IDA can set up any series of test situations or conditions and thus test portions of the application. For example, if a program were written to provide batch updates to one database based on the contents of another, IDA could load data to test the program. Situations that are not likely to occur in the real world can be created, and the program's response to them can be tested. As an application is used in a production environment, IDA could also be used periodically to verify the consistency and integrity of the database in an ad hoc fashion.

DDLP. The commands that specify db_VISTA records, sets, and files (often called the schema statements) are compiled by the DDLP (database definition language processor) utility. The resulting database header and data dictionary files are used by C programs that access the database and by db_VISTA runtime routines (see figure 1).

Input to this utility is the schema file containing the DDL commands for the database, and output are the database header (*filename*.H) and data dictionary files (*filename*.DBD). The DDLP utility reports any schema errors and optionally creates a file structure report and a cross-reference report. The file structure report lists the file ID, size of record slots within the file, number of record slots available per page, and the number of unused bytes per page for each file in the database.

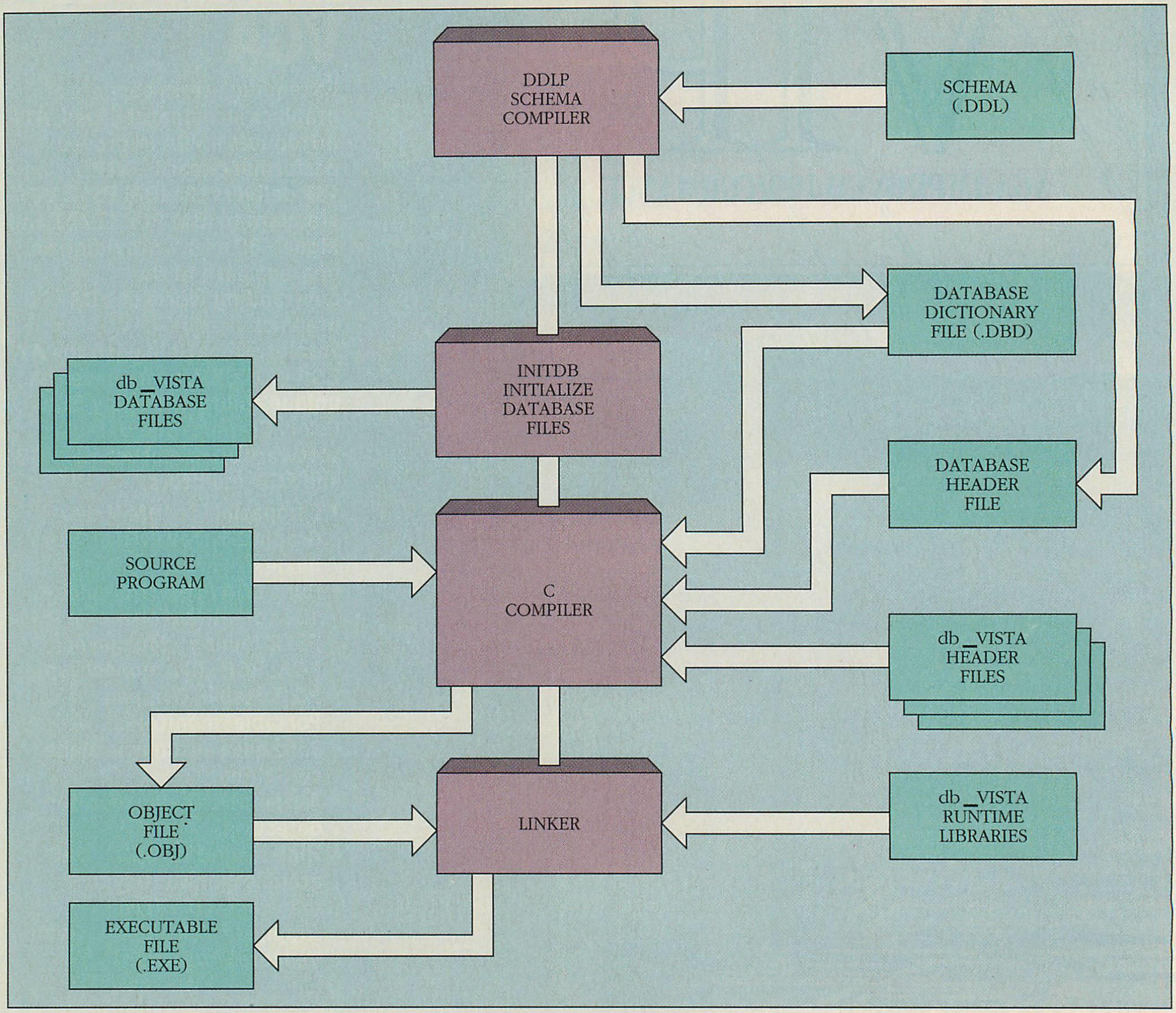
The DDLP utility also lists the record ID, number of owner and member set pointers, size of the record, and number of bytes per slot unused for each record in the database. The cross-reference report lists each record and alphabetically lists each field and set, with its type and the line numbers within the .DDL file where the entity is referenced.

PRDBD. An output from the DDLP utility is the dictionary file (.DBD) for the database specified in the schema. The PRDBD utility displays the contents of this data dictionary and reports default page sizes for each file, along with file, record, set, field, and key information. Developers can use this information to speed up the program or identify areas where disk space can be conserved. The PRDBD utility prints all the table entries in the dictionary file for the database specified.

INITDB. Once the schema has been compiled with DDLP, the resulting database files can be prepared for importing and entering data with the INITDB utility, which initializes db_VISTA data files. This utility creates a page zero for each file in the schema, initializes the date- and time-created fields and the pointers in the system record, and clears the file contents. The program normally prompts the user for confirmation if the database files already exist, but an option lets the user override this prompt, allowing the program to be run from within a batch file.

DBCHECK. The DBCHECK utility verifies the location and key values of each record and key in the db_VISTA database files. An option checks the set consistency, including member and owner record types and membership counts at the owner record. This ensures that all pointers contain valid database addresses. The utility identifies corrupt data; for each error found, the record type and address are displayed in the report. Another option checks only specific files in the database.

FIGURE 1: ***Database Generation and Access***



Database generation and access is a multistep process consisting of compiling the schema using the DDL processor, compiling and linking the C programs that access the database with the db_VISTA runtime libraries, and executing the programs.

KEYBUILD. When new keys are to be added or removed from an existing database or when DBCHECK identifies key files as corrupt, the KEYBUILD program can create the key files from data in a db_VISTA database. This utility is useful when structural modifications are made to existing db_VISTA databases involving key changes in any way. KEYBUILD makes adding keys to existing database records easy and fast.

DCHAIN. After a db_VISTA database has been used for several months and many records have been deleted, the delete chain may need to be re-sorted to ensure that new records use slots vacated by deleted records. The DCHAIN utility sorts the deleted record slots by database address, thus allowing new records created to be assigned lower database addresses. This utility also allows for sorting of the delete chain in all database files or in files specified by the user.

KEYDUMP. To help the developer better understand how the db_VISTA key files are organized, a KEYDUMP utility displays a formatted dump of the key files for the database or for just the key files specified. Key values can be printed in either decimal or hexadecimal numbers, and the report can be redirected to a printer.

The output from this program includes the contents of the zero page and all other pages in the key file. The information displayed includes page size, slot size, number of slots per page, and the contents of each slot on the page displayed. Using this information, the user can walk through a key file viewing each index entry contained within the file.

DATDUMP. To go along with KEYDUMP, DATDUMP is a similar program that displays a formatted dump of the data files in a database. The fields of the database are displayed, along with record header information, which includes record ID, address, and lock bit settings. This program also helps the developer better understand the inner workings of db_VISTA.

The DATDUMP program includes options to select only certain data files in the database, to print the values in hexadecimal format, to display header information, and to select record dumps for only a specified slot number in the file. The information displayed

for each slot includes the record ID, the database address (file number and slot number), and all set pointer values in which the record participates, as well as the actual contents of the data contained in the record.

DBIMP. To import data into a db_VISTA database, the DBIMP utility provides rudimentary load capabilities from ASCII files. This program allows import syntax commands to be created that can specify how data are to be stored in a db_VISTA database. The utility is limited, however, to the extent it can load records that are involved in set relationships and that have no key fields associated with them. For example, records that are stored through a set must be loaded together with their owner records if they are to be stored on the same page.

Furthermore, the user cannot specify searching for an owner record using a compound key. The number of owner records to be stored on a page cannot be limited within db_VISTA or DBIMP. This causes pages meant to hold more than one record to become filled with a single record, thus impairing performance of the database and defeating the purpose of having owner and member records stored together in an application.

DBEXP. The real power of the DBIMP program comes into play when it is used with the export program, DBEXP. This provides an automated means of unloading and reloading existing db_VISTA data when a restructure is

db_VISTA'S NETWORK DATA MODEL

At the heart of db_VISTA is the network data model, which manages data much differently from the prevalent relational data model. In the relational model, the software establishes and maintains relationships through key fields. The network data model represents relationships through linked lists or pointers between records. Database addresses are never changed throughout the life of the record, preventing repetition of data.

Raima's db_VISTA implements the network model using the linked-list concept, where database addresses comprise the assigned file number and the slot number (see figure). In db_VISTA, all databases must be assigned a unique name and all files assigned a unique file number before compilation. Within the database file, records are assigned unique record numbers and the fields that make up the record are C data types in the database header file.

The db_VISTA program has two file types—*data files*, which contain record occurrences, and *key files*, which contain the binary tree nodes for the keys used. A data file could be a customer list and its inventory information. Each data file consists of fixed-sized pages. The system page, called page zero, includes pointers to the start of the delete chain and to the free space. It has other pointers and internal structures db_VISTA uses and time-stamp and database creation information for the database file.

Nonsystem pages within data files are assigned a fixed number of *slots*, physical locations of records within a page. Slots have a fixed length the size of the largest record to be stored in the file plus any system overhead. A number of *fields*—or *data elements*—constitute each record. Together, file, record, set, and field or element definitions make up a *schema*, which defines the database. A database definition language (DDL) processor compiles the schema to create the database's structure.

The db_VISTA routines assign a unique number to each slot and that number becomes part of the address of the record. File numbers can range from 0 to 255 and slot numbers can range from 1 to 16 million for each data file. These restraints allow the developer to create very large databases consisting of many data and key files and millions of records.

Network relationships are known as *sets*; each set has a *set owner* and one or many *set members*. Optionally, the sets can be ordered or sorted based on values in a field or fields on the member records. In db_VISTA, sets are linked lists containing pointers to the members and owner of the set. Set members have a list comprising a pointer to the set owner record, the previous member record, and the next member record in the set. Owner records have a pointer to the first and last members of the set. Sets are also assigned unique numbers, and db_VISTA keeps tables of the structure of the database in the *data dictionary* file. As an example, a set may exist between a Customer and the orders placed by the Customer.

Record type refers to the unique record described in the schema. Continuing the example, the record type "Customer" would refer to Customer data. *Record occurrence* refers to the physical record stored in the database for a customer. A record occurrence is synonymous with a row of a table in the relational model.

Each record has a storage or location mode, which can be by key, by set, or sequentially within the file. Fields or data elements make up the contents of these records, which are assigned to database files. Database records consist of fields; when a field identifies the record or accesses it directly, it is called a *key field*.

Any field in a db_VISTA record can be designated as a key field, which can be set up as structured or compound. Key fields can be designated as unique, allowing only one occurrence of the key value in the index; or as not unique, allowing successive duplicate values following the initial key in the index. The Customer Name can be a key field, and the Customer ID Number can be a unique key field, because multiple customers can have the same name, but not the same Customer ID number.

The db_VISTA key files are also made up of pages consisting of b-tree nodes. Key files must be specified along with data files in the db_VISTA schema, and must be given unique names. Multiple keys can be placed in a single file or different files can be set up for each key in a database.

The actual structure of a key file is similar to the data file. A db_VISTA key file comprises sequential pages with a system page used as with the data file. Page zero consists of a pointer to the delete chain and also a pointer to the free space in the key file. B-tree nodes are stored in each nonzero page in the key file, with each node containing the last update time, the number of slots left in the node, the contents of the index, and a pointer to orphan nodes.

The binary-tree structure is implemented as a linked list, with each entry made up of a pointer to the child node, the key ID, the key value itself, and the database address of the record that contains the key value. The child node contains key values that are less than the value in the current node. The db_VISTA program will split nodes when required, but Raima recommends that developers desig-

required. For example, a client receives an application but decides that some of the data on one record are not needed and another record must be expanded to include new data.

If db_REVISE is not available, the existing data in the db_VISTA files could be exported using DBEXP, the record structures modified as needed, the database initialized using INITDB, and the data reloaded using DBIMP and some import specifications. Although DBEXP and DBIMP can unload and reload existing data in a db_VISTA database, DBIMP is limited in how records can be loaded together with other records. The import utility does not, for example, find an owner record with a compound key. In this case, db_REVISE would be required if any changes needed to be made after a database was installed.

Together, these utilities can maintain a db_VISTA application. The DBEXP component would be especially helpful if data are to be transferred from db_VISTA to dBASE or to a word processor and db_QUERY is not available. The db_QUERY utility outputs an ASCII file and provides more selection capability for exporting than does DBEXP. In addition, db_QUERY is easier to use than DBEXP, but it is not included with the db_VISTA purchase.

FIELDING RECORDS

At the logical level, the editorial database consists of three basic record types: Authors, Articles, and Issues. The

nate a key-file page size that is large enough so that the total number of levels in the binary tree does not exceed four. This recommendation is made in order to prevent multilevel searches for binary-tree nodes and to help improve overall system performance.

Database addresses are simply the file number (1 byte) and the slot number (3 bytes) for the record specified. Each set in which a record participates as a member thus adds 12 bytes to the size of the slot into which the record will be placed. At the owner record, the linked list is made up of the number of members in the set, with pointers to the first and last members in the set. The owner record thus has 12 bytes added to its slot size for each set in which it participates as an owner record. The db_VISTA package also provides for indexes on records and access by key value, but these structures are not implemented as sets in the db_VISTA database.

The page sizes for key files and data files in db_VISTA can greatly affect the database performance and the amount of space required. Several factors come into play. The basic premise of the network database is that related records are stored physically close to one another to save I/O overhead. Thus, owner records could be stored on the same page as member records in the set and potentially only one read would be required to access all records in the set.

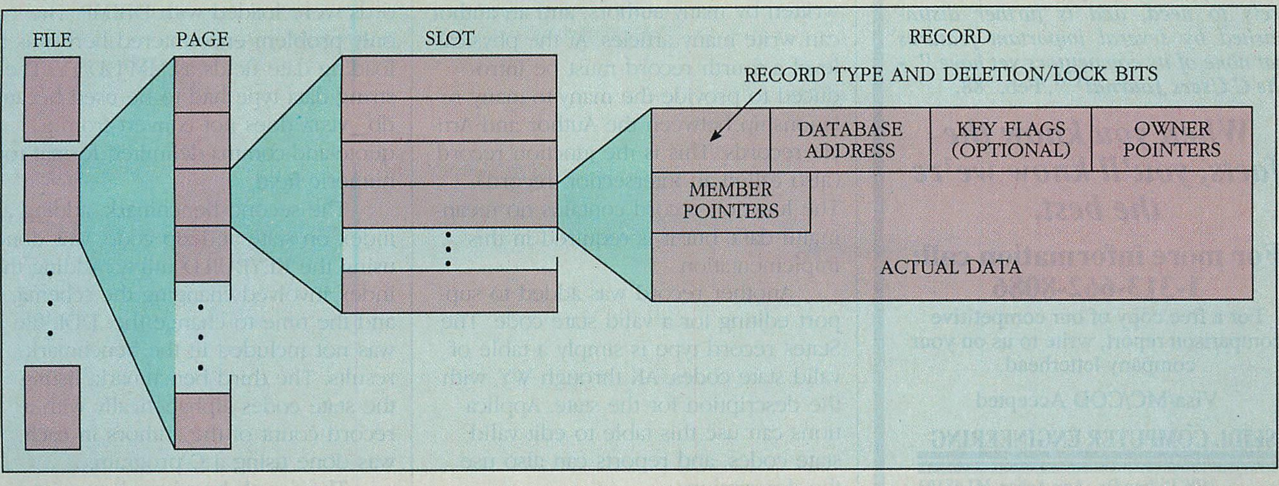
However, this simplistic approach often is difficult to implement. With fixed-sized slots in db_VISTA files, placing large owner records in the same file as smaller member records can waste a significant amount of disk space depending on the average number of members in the set. Yet if member records are placed into another file with a smaller page size, more I/O will be needed to access the member records when the owner record is read. The trade-off of disk space versus access time must be considered when designing the database to evaluate the best combination for the application.

Another design issue is using indexes on records. Each indexed field requires a db_VISTA key file and the overhead at runtime to support the entry and modification of the index entries. Disk space also comes into play when a large number of index entries are required. For example, suppose an application contains a last name, first name, and middle initial for each entry on a record. To support an index on 26 characters, say 15 for last name, 10 for first, and 1 for middle initial, the index entry would be around 35 bytes.

When only 1,000 entries are involved, the decision is simple. But when more than 1 million entries are needed, the decision becomes more difficult. Estimating the number of entries required often can make the design decisions more appropriate for the application.

—Andrew Topper

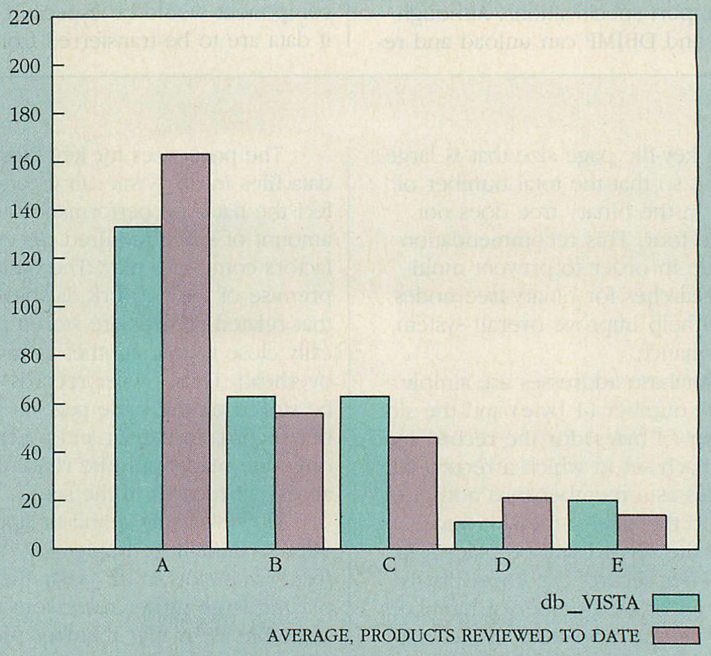
FIGURE: ***db_VISTA Database Structure***



db_VISTA databases consist of files that are made up of pages composed of a fixed number of fixed-size slots. Each slot must be large enough to store the largest record in the database. Records consist of the actual data plus pointers and flags.

FIGURE 2: ***Benchmark Results***

BENCHMARK TASK	TIME
A. Add 900 records to an empty database table	133
B. Index table on two fields (7 bytes)	63
C. Document and tally codes from one column	63
D. Mass change of one column (28 rows of 900)	11
E. Extract selected records to create a text file	20



All times are in seconds.

All benchmarks were run on an IBM PC/AT (6 MHz) with 640KB memory. The tests were run in an 8MB partition on a CMI 20MB hard disk under DOS 3.0.

db_VISTA was faster than average when adding 900 records to the database, but except for benchmark D, it exhibited only average performance. This benchmark used a C program written to use the index on states and zip codes; benchmark E created an ASCII file of California authors sorted by zip code.

relationships identified are: an issue contains many articles; an article can be written by many authors; and an author can write many articles. At the physical level a fourth record must be introduced to provide the many-to-many relationship between the Author and Article records. This is the junction record (also called an intersection record). The Junction record contains no meaningful data, but it is required in this implementation.

Another record was added to support editing for a valid state code. The States record type is simply a table of valid state codes, AK through WY, with the description for the state. Applications can use this table to edit valid state codes, and reports can also use the description.

The benchmarks for the editorial database (see figure 2) were performed with db_VISTA tools and, in general, presented few problems. Author records were loaded with DBIMP. The only problem encountered here was in loading date fields as MM/DD/YY. The string data type had to be used because db_VISTA does not convert from a quote-and-comma-delimited format to a numeric field.

The second benchmark, adding an index on state and zip code, was done using the KEYBUILD utility. Adding the index involved changing the schema, and the time to change the .DDL file was not included in the benchmark results. The third benchmark, listing the state codes alphabetically with a record count of the authors in each, was done using a C program.

The fourth benchmark required writing a C program to modify each of the Author records that contain the

state code CO to CL. This program, which was written to use the index on states and zip codes, ran very quickly. The db_QUERY program generated the fifth benchmark, which involved creating an ASCII file of all the authors in California sorted by zip code. In general, the benchmarks required no special handling when db_VISTA and db_QUERY were tested.

The db_VISTA package is an excellent implementation of the network model database, and it can be suitable for large, complex applications development. The popularity of C and its portability to many operating environments make db_VISTA appealing to a

Although db_VISTA deserves praise, Raima could add more utilities, including a cross-reference utility and an analyzer.

wide range of developers and programmers. With the multiuser and LAN support, developers can use db_VISTA for almost any application requiring excellent performance and transaction-oriented processing.

Together with db_QUERY and db_REVISE, db_VISTA provides all the tools needed to support the complex applications being developed for the 80286- and 80386-based computers. The single-user version, coupled with screen-handling products such as Windows for C (see Product Watch, Paul Schauble and Rick Cook, December 1987, p. 197), delivers a complete development environment suitable for smaller applications.

Even though db_VISTA deserves praise, Raima could improve the product by adding more utilities. In particular, the program needs a cross-reference utility to map C source code to db_VISTA database files, records, sets, and even fields; this would be a helpful addition for maintenance. This information is critical when determining how a change may affect an existing application. To meet this demand, such a utility could also generate a list of those programs affected by a change to a db_VISTA schema.

As a network model database ages, an analyzer utility is needed to map set member locations, the number of deleted records in the delete chain, and the percentage of empty pages in the database. This utility could pinpoint problems in the structure of a database and help address areas of the original database design that may have been incorrect. An analyzer utility could also identify areas in the database that may need investigation to improve performance and reduce space.

The information available from the KEYDUMP and the DATDUMP utilities helps identify these problems, but when the number of pages in a database exceeds 20 or 30, manual analysis can become overwhelming. An analyzer utility has become common on mainframe data managers that use the network data model.

Also, db_QUERY can be an inefficient reporting tool for end users. Although a developer could try to identify and develop all the major reports required by the user and .QRF files could be established for each report, this is rarely possible from a practical standpoint. Most users cannot accurately identify their exact needs before the application is installed. Therefore, a good application provides for ad hoc reporting that can be used when necessary. db_VISTA should provide a more convenient reporting capability than the current version of db_QUERY. For example, db_QUERY 2.0 has no on-line help function.

However, db_VISTA's features overcome its faults. The product provides the development tools necessary to construct complex business database applications. Deciding whether to develop with db_VISTA rests on how two factors would affect your development effort. First, consider how the network data model would fit into applications. The model's file structure may better suit certain databases than the structure of the relational model—although bear in mind that the network data model requires extra effort in order to modify existing applications.

Second, db_VISTA is one of the few packages that offers the portability of the C language, giving database developers abilities other software developers have enjoyed for some time. With the benefits of both the network data model and C, Raima's db_VISTA proves it has the power to compete in a marketplace overstuffed with relational data managers.

Andrew Topper is president of Foresite Systems, a Lansing, Michigan, consulting firm specializing in database and application design for LAN and multiuser systems.

No matter which Ethernet LAN you decide on, fiber will make it decidedly better.

Choose proven Ethernet network interface controllers like those from 3Com, Novell or Western Digital. Use proven Ethernet software like Novell Netware. Then come to Versitron for compatible fiber optic transceivers and star hub concentrators that let your LAN communicate on fiber optic cable instead of coax or twisted pair.

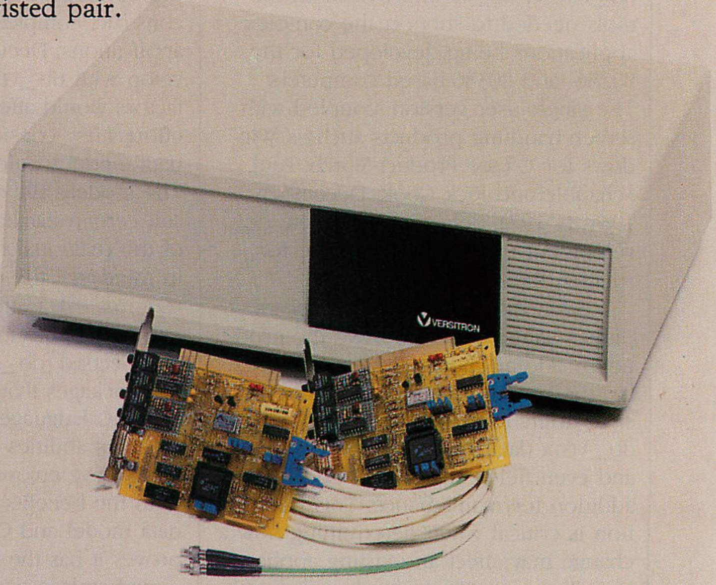
With fiber optic cable, your LAN will be reliable. The medium can't be disturbed by noise from adjacent heavy equipment wiring. It won't suffer from ground currents. It won't even be bothered by lightning.

Since fiber does not carry electrical current, it won't radiate RFI, and it's very difficult to tap fiber without being detected. Fiber is lighter and more flexible so it is much easier to install. You can transmit up to 2000 meters between nodes compared to the 300 to 1000 meter range of coax or even shorter range of twisted pair.

With Versitron transceivers and concentrators, your Ethernet can be a combination of bus and star topology to match network geography.

If you've never used fiber before, you might be interested in our starter kit. For just $995, you can put 2 nodes of your Ethernet LAN on fiber.

If your network's performance is important, there really is nothing to decide. Fiber is the right medium. And only Versitron offers fiber with the most popular Ethernet network controllers and software. Call us today at (301) 497-8600.



VERSITRON
A Keene Corporation Company

9005-8 Junction Drive Annapolis Junction, MD 20701 • Telephone (301)497-8600 Telex 4972387

CIRCLE NO. 134 ON READER SERVICE CARD

PRODUCT WATCH

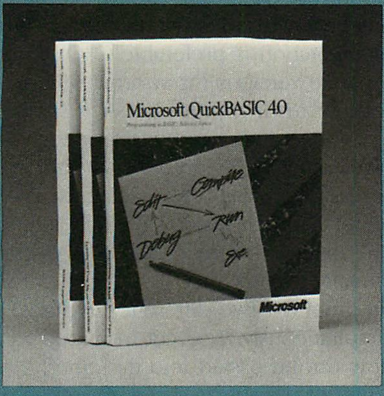
Reviews and Updates

QUICKBASIC 4.0
Microsoft Corporation

MICROCACHE 4.37
Microcosm Research

QUICKBASIC 4.0
Microsoft Corporation
16011 N.E. 36th Way
P.O. Box 97017
Redmond, WA 98073-9717
800/426-9400; 206/882-8080

PRICE: $99



CIRCLE 332 ON READER SERVICE CARD

As a result of intense competition among compiler vendors, BASIC programmers are blessed with affordable, high-quality compilers and development environments from both Borland International and Microsoft. A recent comparison of Borland's Turbo BASIC and Microsoft's QuickBASIC version 3.0 ("The BASIC Face-Off," Justin Crom, September 1987, p. 136) concluded that the real winner in the contest was the programmer. Microsoft's QuickBASIC version 4.0 (QB4) is a striking improvement over QB3, tipping the contest in favor of Microsoft. QB4 is not just another rework of previous versions; instead, it is a markedly different offering.

Most significantly, QB4 effectively combines aspects of both an interpreter and a conventional compiler. It retains interpreted BASIC's direct mode as well as the ability to run a program without going through a compilation step. The user may interrupt a running program, change the source code, and resume execution with data intact from the point of interruption: yet, the code executes with the speed expected from compiler output.

Microsoft accomplishes this blending by incrementally compiling each line of source code as soon as it is entered. The partially compiled statements are then threaded together at runtime into the completed program. This is done in a manner transparent to the programmer.

To enable swift conversion back and forth between the text source code and the executable code, three program states are used. As program statements are entered, either with the built-in editor or from a text file prepared with an external editor, QB4 checks their syntax and translates them into the *parsed* state. When the program is run for the first time, QB4 converts it to the *symbolic* state by creating a symbol table for the variables. It then immediately progresses to the *threaded* state, which is the executable form.

The threaded p-code is similar to the output of a compiler and executes nearly as fast as fully compiled native code, yet it retains sufficient information about its origin so that the source code may be reconstructed from it for subsequent editing. Unlike traditional pseudocode that represents each atomic operation of the language with a token, the threaded variety represents an operation by an executor address, that is, the address of the procedure that performs the operation. The traditional interpreter uses a massive switch statement to test each token and branch to the appropriate procedure; the threaded interpreter merely loads the next executor address and branches to it without testing. Furthermore, each executor ends with a similar load-and-branch sequence to the next executor. Thus, execution threads its way from procedure to procedure without the intervention of a centralized interpreter routine.

Conventional compilers parse the source code and perform syntax checking only when compilation is invoked. By parsing each line of source code immediately upon entry, QB4 gets a head start on the compilation. The remaining steps to produce executable code proceed quickly: the entire sequence of state changes (from parsed through symbolic to threaded) proceeds about ten times faster than the single-step compilation in QB3.

Microsoft's innovative use of intermediate states speeds up editing and execution after the first running of a program. Because a majority of program changes during an edit session requires only translation from threaded back to symbolic state, editing and rerunning a program with QB4 avoids complete recompilation of the code. Only a few changes require conversion completely back to the parsed state (for example, changes to COMMON, SHARED or DEF statements).

Changing between symbolic and threaded states in either direction proceeds about 50 percent faster than the change between parsed and threaded. Following a typical edit session, program recompilation after initial execution proceeds more than twice as fast as the first compilation. With conventional compilers, each compilation takes the same amount of time, and compilation delays can become frustrating, especially if brought about by a small change to the source program.

Threaded p-code is used only for interpretive execution within the QB4 development environment. As with previous versions of QuickBASIC, a program also may be compiled to a standalone .EXE file, with library routines either incorporated within it or loaded at runtime. However, because interpreted and stand-alone forms consist of different code, the results produced by

TABLE: *Benchmark Results*

	MICROSOFT		BORLAND
PRODUCT	QuickBASIC		Turbo BASIC
VERSION	4.0	3.0	1.00c
PRICE	$99	$99	$99
SIEVE (per iteration)			
Execution time in memory	2.3	0.3	0.4
Execution time, .EXE file	0.3	0.3	0.4
Size of stand-alone .EXE file	25.9	28.1	22.2
MULDIV (50,000 iterations)			
Execution time in memory	27.2	20.1	23.0
Execution time, .EXE file	20.0	20.1	23.0
Size of stand-alone .EXE file	27.1	30.4	23.5
HAT			
Execution time in memory	128.7	107.3	181.6
Execution time, .EXE file	101.6	107.3	181.6
Size of stand-alone .EXE file	46.9	38.7	25.4
FILEIO (30KB)			
Execution time in memory	128.4	118.9	119.1
Execution time, .EXE file	128.2	118.9	119.1
Size of stand-alone .EXE file	31.6	32.8	24.1
SAVAGE (25,000 iterations)			
Execution time in memory	37.8[a]	36.1	29.5
Execution time, .EXE file	29.7	36.1	29.5
Size of stand-alone .EXE file	27.7	32.2	22.8
SCRNTEST (100 iterations)			
Execution time in memory	5.8	93.5	22.8
Execution time, .EXE file	5.7	92.7	7.8
Size of stand-alone .EXE file	27.3	31.1	21.8
DRAWTEST (50 iterations)			
Execution time in memory	17.5	17.4	15.3
Execution time, .EXE file	17.4	17.4	15.3
Size of stand-alone .EXE file	45.8	43.0	28.6
NUMERICAL ERROR RATING[b]			
In memory	0.76	0.79	0.80
.EXE file	0.83	0.79	0.80
BIGTEST, 2,000 lines			
Load time, text form	7.4	0.5	0.8
Save time, text form	4.5	0.8	0.8
Load time, parsed form	1.4	N/A	N/A
Save time, parsed form	1.0	N/A	N/A
Compile to memory and run	2.3	20.2	9.8
Recompile to memory and run	1.2	20.2	9.8

Times in seconds, sizes in kilobytes.
[a] *Calculates wrong result; see text.* [b] *Smaller values indicate higher numeric accuracy.*

Although compiling in memory to p-code produces slightly slower programs, the compilation time, especially after minor changes are made, is greatly increased.

a program in one environment may not be the same as in the other. An example is given later in this review.

While the innovation of threaded p-code is the most outstanding accomplishment embodied in QB4, it differs from its predecessors in other respects. Microsoft has addressed most of the shortcomings of QB3 in comparison with Turbo BASIC's advanced language features. In QB4, subprograms can be recursive, meaning that a function or subroutine can call itself. This is a powerful technique for programming mathematical computations and processing complex data structures. Lower array bounds can be set to any integer value, not just 0 or 1. Like Turbo BASIC, QB4 also supports long (32-bit) integers and binary files. Going beyond the capabilities offered by Turbo BASIC, QB4 introduces a TYPE declaration, analogous to a record in Pascal or a structure in C, that permits grouping of different variable types into a single entity. This simplifies record-oriented file I/O, providing a long-awaited alternative to the awkward FIELD statement.

One characteristic unchanged from QB3 is memory usage. QB4 uses the medium-memory model, allowing unlimited program space but no more than 64KB for all data, including strings and static arrays. Dynamic numeric arrays are allocated outside of the one data segment and may exceed 64KB, both individually and aggregately.

Installation is straightforward and involves copying files from only three disks. QB4 works with or without a math coprocessor, automatically switching between hardware and software floating-point routines depending on the presence or absence of the coprocessor at runtime. Software emulation may be forced, if desired, by setting an environment variable. This implementation is preferable to QB3's separate versions for systems with and without math coprocessors.

QB4's integrated environment owes much to its predecessor, but has several significant enhancements. The editor is much improved over QB3. The main edit screen may be split horizontally to view or edit two parts of a program (or two different programs) simultaneously. Text cut from one window can be pasted into the other. Some of the shortcut keys have been changed to operate more logically; for example, the Ins key toggles between insert and overtype modes and the Del key deletes a single character. Cut-and-paste operations, assigned to these keys in QB3, are now performed by Shift-Ins and Shift-Del. However, this means that the shift key cannot reverse the NumLock state of the numeric keypad.

As program statements are entered, QB4 checks syntax and formats the line by capitalizing BASIC key words and by adding spaces before and after operators. Punctuation that was inadvertently omitted by the programmer is added where required. For example, QB4 will insert semicolons between items in a PRINT statement argument list and will supply closing quotation marks if omitted. QB4 tracks variables and procedure names, maintaining consistency of capitalization. The capitalization of every occurrence of a name is changed when any one occurrence is changed.

Program lines entered with QB4's editor are limited to 255 characters,

and they may not be continued. However, both the editor and the compiler readily can handle longer lines created externally. Also, program files created outside of the QB4 environment may use the QB3 continuation character, the underscore (_), to break up long lines into shorter segments. The underscores are removed and the lines are concatenated when the program is loaded into QB4.

As with QB3, the use of a mouse is a plus but not a requirement. The use of color has been implemented intelligently but with consideration for monochrome systems. A major complaint about QB3 on monochrome systems without a mouse was the occasional difficulty discerning which option had been chosen. With QB4, the chosen option is clearly indicated by gray-scaling.

A minor but irritating bug is present in the file manager. When requested to load a file, QB4 displays an alphabetized list of subdirectories and files that conform to a wild-card pattern (by default, it is *.BAS in the current directory, but it can be changed). The user picks a file to load by pointing to it with the cursor and then clicking the mouse or pressing the return key. The mouse works without problem in all cases, but the return key has no effect when the cursor is on the first file name in the list. This condition can be corrected by moving the cursor around the list and returning to the first entry (providing there is more than one entry). Otherwise, a single file can only be selected with a mouse.

As previously mentioned, test runs within the environment execute from an intermediate code that is produced partly during editing and partly just before execution. No separate compilation step is involved; the user merely chooses RUN from the main menu and the program starts. However, the user also may request compilation to either a .OBJ or .EXE file. As in all previous incarnations of Microsoft's BASIC compilers, the .EXE file may incorporate either its own copy of library routines or use a runtime library.

Compilation to native code is performed by a separate command-line compiler that may be invoked either from within the QB4 environment or directly from DOS. If the creation of a .EXE file is requested from within QB4, the standard Microsoft linker (supplied with the compiler) is automatically invoked after successful compilation. Thus, the need to leave the environment to perform a link step—a major flaw of QB3—is corrected.

Compiling from the DOS prompt might be useful to programmers used to their own text editors through habit or edict; of course, this negates the development speed advantages provided by testing with pseudocode. Another reason for external compilation is the ability to take advantage of compiler options not available from within the environment. This includes storing arrays in row-major order and producing debug information for Microsoft's CodeView, an advanced debugger available with other Microsoft languages, but not provided with QB4.

One of the major additions to QB4 is the set of module-management features. First, multiple modules may be loaded simultaneously into the editor from separate source files. Second, each subprogram within a program file is displayed as a separate module. Although only one or two modules may be viewed at once, switching between them is fast and simple. The VIEW function of the main menu displays an outline of the program, listing the main module and the subprograms in all files currently loaded. From this list,

the user may choose a module to edit by clicking with the mouse or keyboard. This system encourages structured programming and avoids much of the tedious scrolling that is associated with locating and displaying subprograms in other editors.

When a multimodule program is saved, QB4 creates a .MAK file listing all the dependencies. Thereafter, whenever the main program is loaded into the editor, QB4 automatically loads all of its modules and keeps track of which ones need to be recompiled when the program is rebuilt.

For programs consisting of more than one module, QB4 automatically inserts subprogram declaration statements and reorders the subprograms alphabetically by name when the file is saved, run, or compiled. Each declaration specifies the name of a subprogram, whether it is a function or subroutine, and the number and types of arguments. Like function prototypes in C, the declarations are optional and are used to check the validity of function calls—a significant and useful extension to the BASIC language.

The automatic modification of a source file becomes an issue when loading files are created externally. When compiling from within the environment, QB4 insists on inserting the declarations and saving the new source code. The user is asked for permission to overwrite the source file. If the user declines to give that permission, the compilation is not performed. This seems excessively restrictive because the compiler does not require declarations. Programs without declarations can be compiled by invoking the compiler from the command line.

QB4 uses two types of object-module libraries. Modules to be linked into .EXE files are gathered into .LIB files and are created and maintained by Microsoft's LIB.EXE (provided with QB4). Modules that are called from within the environment are gathered into .QLB (Quick Library) files. Both types of libraries may be created from within the programming environment or from the command line.

One of the most impressive features of QB3 is the debugging provision's animated trace mode in which successive statements in the source code are highlighted as they are executed. QB4 retains this feature and adds other capabilities to make the debugging chore less odious. Watchpoints may be specified to suspend program execution when variables reach target values. A HISTORY option permits the last 20 statements executed to be examined in either forward or reverse sequence. The capabilities of the integrated debugger are a subset of those available in the CodeView debugger. In the case that the integrated debugger proves insufficient, the command-line compiler can produce .EXE files that support all of the advanced features of CodeView.

With QB3, the debug trace takes place in a small window that does not obscure the output screen. Unfortunately, QB4 does not have this feature. Tracing through programs that output to the screen causes some flickering when QB4 switches back and forth between the environment's screen showing the animated trace and the program's output screen.

In QB3, HELP consists of one screen of commands. With QB4, the HELP facility has been greatly expanded and now provides context-sensitive help. When editing source code, placing the cursor on a key word and pressing Shift-F1 shows a synopsis of the syntax for the statement.

The documentation is excellent, on its own and in comparison with previ-

ous versions. The two-inch thick, 670-page manual of QB3 has been replaced with three volumes totaling over 1,300 pages. The first volume is a user's guide describing installation and operation of the integrated environment; it also describes the specifics of this dialect of the BASIC language. The second volume is a tutorial on BASIC programming, describing the various language statements in functional groups. The third is a BASIC language reference with all statements and functions listed alphabetically. All three volumes are logically organized and well-written. The section on separate compilation and linking, a weak spot in previous versions, has been doubled. Although the instructions do not assume a high level of familiarity with the concepts and procedures of compiling and linking, the tone is never patronizing. Novice and expert alike will find the documentation instructive and informative.

The performance of QB4 was tested with a suite of programs previously published with various reviews of BASIC interpreters and compilers. For SIEVE, MULDIV, FILEIO, and HAT, see "Six New Shapes of BASIC," Ted Mirecki, June 1986, p. 52; for SAVAGE, SCRNTEST, and DRAWTEST, see "BASIC Face-off," Justin Crom, September 1987, p. 136; for ACCURACY, see "Measuring Numeric Accuracy," Jim Roberts, January 1988, p. 142. All of these programs are available for downloading on PCTECHline.

As shown in table 1, QB4 creates stand-alone .EXE files that are, in most cases, slightly more compact than those generated by QB3, but are not as small as Turbo BASIC's. Except for the dramatic improvement in speed of text output (as measured by the SCRNTEST program), execution speed of .EXE files is only marginally improved over that of QB3, which in turn is better than Turbo BASIC's speed. As expected, the execution speed of programs compiled to threaded p-code in memory is somewhat slower. However, this form is meant for development, not production use, and the advantages it brings to the development cycle far outweigh the minor speed penalty.

These advantages are demonstrated by the timings for the BIGTEST program. This is a large program consisting of the following statements:

```
A = 1
A = A + 1
(above repeated 2,000 times)
PRINT A
```

QB4 can compile to memory and run this program more than four times faster than Turbo BASIC, the previous champion of compilation speed. The reason for this increase is that part of the compilation (conversion to the parsed state) occurs when the program is loaded, as shown by the longer load times of QB4. Although the total time of loading, compiling, and running is greater in QB4 than in Turbo BASIC, the trade-off between load time and compile time is reasonable. In an integrated development environment, compilation and execution occur much more frequently than loading does. Furthermore, QB4 can load and save files in parsed form (analogous to the tokenized format of the BASICA interpreter) that, for all practical purposes, eliminates the speed advantage of the Turbo BASIC file handler.

The advantage of QB4's incremental compilation is even more dramatic when rerunning the program after minor modifications. After changing one line in the program, QB4 can recompile and run in half the time of the initial execution, while Turbo BASIC repeats the same process all over again. The cumulative time saved in the development cycle of a major program can be substantial.

Of course, with every advantage comes a concession. In this case, it is that the p-code executed during the development phase is not identical to that in the .EXE file created for the production version of the program. Therefore, the results obtained during testing may not be identical to those from the final form. This is dramatically illustrated by the results from the SAVAGE program, which in p-code form produces a result of 4,099 instead of the correct value of 25,000. According to Microsoft, p-code uses 8-byte temporary real variables, while .EXE code uses 10-byte variables. However, this difference does not explain why the error rating measured by the ACCURACY program is better for p-code than for .EXE files (a lower error rating indicates better accuracy).

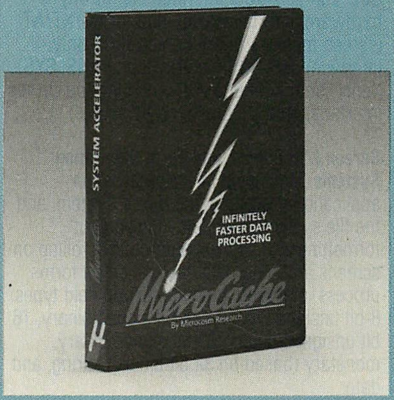
Whereas QB3 is a competent compiler, it shows evidence of being a patched-up version of QB2. Microsoft apparently needed a stop-gap measure to parry Borland's Turbo BASIC while QB4 was being developed. The contest between Turbo BASIC and QB3 was declared more or less a draw, with Borland's offering being more suited for less experienced programmers. QB3, with its provision for separate compilation and linking, was aimed toward the more experienced programmers. QB4 overcomes most of the shortcomings of its predecessor. Improvements are made in performance, ease of use, documentation and, above all, speed of development. In turn, these improvements make QB4 the premier BASIC and the logical choice for the novice and professional alike.

—JUSTIN CROM

MICROCACHE 4.37

Microcosm Research
Distributed by Az-Tech Software Inc.
305 E. Franklin
Richmond, MO 64085
800/227-0644; 816/776-2700

PRICE: $69, without print buffer
$95, with print buffer



CIRCLE 334 ON READER SERVICE CARD

Disk caches speed access to disk-based data. Such utilities are becoming increasingly popular; a caching utility is even included with IBM PS/2s. "The Cache Factor" (Maxine Fontana, August 1987, p. 168) discussed the operation of caches, and the Product Watch column of the same issue (p. 211) discussed Multisoft's Super PC-Kwik. Another such package is Microcosm Research's MicroCache 4.37.

Like RAM disks, caching utilities can help certain applications run significantly faster because they store data and files in memory. Unlike RAM disks, however, cache utilities are "smarter," more flexible, less vulnerable, and generally transparent to an application. A cache facility maintains a pool of buffers (cache memory) that it uses to store recently or frequently used blocks of disk data.

The cache utility intercepts read and write requests to a target disk, and checks to see if the requested data are

TABLE: ***MicroCache Performance Tests***

	WITHOUT CACHE[a]	128KB CACHE[b]
Sequential file creation	7.06	7.05
Sequential file write	9.67	10.46
Sequential file read	6.01	6.08
Random file write	15.68	13.73
Random file read	12.19	3.26
Word spell check (average)	41.61	31.43

All times are in seconds.
[a] *BUFFERS = 10* [b] *Write buffering disabled*

Due to the fundamental nature of caching mechanisms, the greatest increase in performance is normally obtained by randomly rather than sequentially accessing data stored on a cached disk. Unless a sequential file is read multiple times using a large cache buffer, little, if any, performance benefit will be realized. In certain cases a degradation in performance can occur due to the overhead of the cache software and the fact that additional physical disk accesses may be required.

in its cache memory. If not, it physically reads from or writes to the disk. For write operations, caching software commonly provides a write-through capability that forces write operations to be physically written to a target disk immediately; this helps ensure the integrity of the data in the event of system failure. MicroCache is a write-through cache, but optionally supports a write-buffering feature.

MicroCache runs as a terminate-and-stay-resident (TSR) program; the program provides no method of removing itself from memory, short of rebooting. In addition to the disk cache and buffered disk write facilities, MicroCache provides three other functions: a print spooler, a RAM disk, and a facility to speed screen I/O. Microcosm Research includes the RAM disk for comparative purposes and does not recommend its use.

Although MicroCache purports to speed screen I/O for certain screen functions, tests showed the benefits to be insignificant. Without the cache software installed, it took 3.50 seconds to write 10 screens of 24 lines each to the display using video BIOS calls. The same test with the cache software installed took 3.47 seconds.

To use its print-spooling function, MicroCache allocates a portion of its cache memory to spool print requests. This speeds print applications considerably. Control of the print, spooling facility is provided via hot-key access to a windowed menu that permits the user to pause printing, flush the print from the buffer, or send a form-feed request to the printer. The spool contents also can be redirected to any of four printers (two parallel and two serial) or, with the help of nonresident utilities supplied with the package, can be sent to a disk file.

To guarantee fastest access, one of the support utilities locks files and disk directories into cache memory; a companion utility unlocks those files and directories. With the exception of the support utilities and the RAM disk driver, all the functions are integrated into a single package.

An installation utility toggles the print spooler and RAM disk, sets the default cache size, allocates more RAM from extended or expanded memory, and toggles caching and write buffering for each disk drive. It also allocates memory to the RAM disk (an allocation of 0 disables the RAM disk). The only parameter that can be modified at run time is the amount of conventional memory dedicated to the cache.

MicroCache loads into conventional memory. Its buffer can be installed completely into standard extended or expanded memory or divided between them. The utility uses a minimum of 70KB and can use up to 4MB. It can cache both hard-disk and diskette drives; up to 16 drives can be cached. The utility uses "least recently used" (LRU) and "least frequently used" (LFU) algorithms to determine which disk sectors to preserve in its cache memory and which to allow to be flushed to disk.

When the write buffering function is enabled, writes to a target disk are held pending in the cache's memory until the processor is not busy; MicroCache then writes the data to disk. Data will be lost if normal system operation is compromised before the writes are actually posted to disk. If write buffer-

ing is used with a diskette, MicroCache attempts to recognize when a diskette has been changed and traps, requesting confirmation that the proper disk is in the drive. On AT-class machines, only disk changes in the 1.2MB diskette drive are recognized and trapped; Microcosm Research does not recommend caching for drives in which media change cannot be detected.

As expected, MicroCache permits software to access data faster. Two tests measured the program's performance. The first measured the time needed to create, read, and write a 265KB file. These operations were done both randomly and sequentially using two buffer sizes (512 bytes and 4KB). The other measured the time needed to check the spelling of two Microsoft Word documents. These files were 5,760 and 25,856 bytes long, respectively, and contained no spelling errors.

Table 1 gives a summary of the performance test results. The write buffering function was disabled for the tests. In general, MicroCache was faster performing reads rather than writes and accessing data randomly rather than sequentially. The results were predictable given the nature of cache operation. The effectiveness of caching is difficult to generalize without considering the application environment.

Obviously, compute-bound operations will not benefit greatly from a cache, because disk use is minimal. Even some disk-intensive applications will not benefit from caching. For example, if the cache is too small, the chance of the cache containing the requested data is also small, and time must be spent managing the cache as well as retrieving the data.

If a file is read sequentially, from beginning to end, no block is going to be read a second time, so caching only adds overhead to the application.

For applications that are interactive or those that do not access read data sequentially, the probability that requested data will be found in the cache's buffer pool is greater; these applications can benefit directly from the MicroCache's capabilities. Most applications that perform writes, sequential or otherwise, will benefit if the write buffering option is enabled.

Overall, MicroCache is a good product that offers a usable caching facility along with added productivity features. The integration of print spooling with the cache makes the software particularly attractive.

—PHILIP N. HISLEY

OS/2 UNIX

New VCScreen 2.0

■ WINDOWS ■ DATA ENTRY ■ MENUS ■ HELP MANAGEMENT ■ TEXT EDITING ■

Vitamin C

PROFESSIONAL C LANGUAGE FUNCTION LIBRARY

- ❑ Multiple bullet proof overlapping windows
- ❑ Easy single field or full screen data entry
- ❑ Unlimited data validation
- ❑ Context sensitive help manager
- ❑ Menus like Lotus & Mac
- ❑ Programmable keyboard handler
- ❑ Text editor routines
- ❑ Printer output routines

- ❑ 30 day money back guarantee
- ❑ No royalties or runtime fees on applications
- ❑ Complete library source code included FREE
- ❑ FREE technical support
- ❑ FREE BBS at (214)418-0059
- ❑ Supports Microsoft 5, Quick C, Turbo C, Lattice and others
- ❑ Optional screen painter/generator

Better Applications In Less Time

Fast, flexible, versatile, reliable. Just some of the reasons why serious programmers use Vitamin C in their most important projects. They know using Vitamin C means lightning fast displays, a responsive user interface, professionally crafted C code, and a commitment to technical support.

High level functions provide maximum speed and productivity. Extended versions of these same routines add flexible control over specific details when necessary.

Generate Code Inter

NEW VERSION!

Speed development even more with VCScreen, our interactive screen painter / code generator. Define windows, boxes, borders, headings, input and output fields. Copy, delete, change, move, even layer objects. Then let VCScreen generate C source code ready to compile and link with the Vitamin C function library.

New features allow creation of multiple windows, menu systems, global variable maintainance, user defined code generation options, and more user configuration options!

Versatile Design Keeps You In Control

Options and possibilities rather than limitations and frustrations mean you're always in control. Our versatile open ended design is full of hooks so you can intercept and plug-in your own control functions to easily customize or add features to most routines.

Easily create windows that pop-up, overlap, zoom, move, scroll, hide, show and resize. You'll choose options for titles, borders, colors, scroll bars, virtual size, and more. You can even access any window any time, even if it's hidden or invisible. That's flexibility.

Sophisticated data entry forms become easy with features like unlimited validation, protected, invisible, and scrolling fields, full color control, single and multiple field input, selection sets, even right-to-left numeric input! And, with the context sensitive help system it's easy to provide field specific or other help messages.

Vitamin C's menus are the perfect framework for any application and feature advanced options such as check marks, unavailable items, blank items and separators.

The keyboard handler routines can redefine key assignments, translate keystrokes, even call a function.

Utility routines for time/date management, background processing, and sending windows to a printer.

Thorough documentation with tutorial and reference sections. Reference databases compatible with the Norton Guides Instant Access Program are also available.

Users And Reviewers Agree

"Picking the best value package is hard... If you're a source code fanatic like me, Vitamin C is preferable. If you need source code, make sure your wallet is wide open or get Vitamin C."
Computer Language, June '87

"Only Vitamin C supports keyboard handlers and keyboard reassignment. Vitamin C provides the most options for menus."
BYTE, October '87

"I trust our review of [Vitamin C] in Computer Language magazine was fair . . . it has become the screen manager package of choice at my firm."
Michale Wilson, Wilsoft, Inc.

OS/2, UNIX and Xenix versions now available. Call for prices and details.

Vitamin C.........$225.00
Includes source. Specify compiler when ordering.

VCScreen..........$149.00
Requires Vitamin C library above.

Reference Database...$50.00
Requires the Norton Guides program sold separately.

Requires IBM PC, XT, AT, PS/2 or compatible. Include UPS shipping: $3 for ground, $6 for 2nd day air, $20 for overnight, $30 if outside U.S. All funds must be in U.S. dollars drawn on a U.S. bank.

ORDER NOW!
(214) 416-6447

creative PROGRAMMING

Box 112097 Carrollton, Tx 75011

TECH NOTEBOOK

1 MEMORY MODELS

2 FINDING FILES

Version 5.0 of Microsoft's Macro Assembler enhances an already excellent product, but even it could stand some improvement. The first item in this month's Tech Notebook is a small file that provides a large degree of convenience in writing code that is independent of the memory model for which it is assembled.

The second item addresses the problem of finding an application's auxiliary files, such as code overlays, configuration specifications, and device drivers. The method used by most applications—searching in each subdirectory named in the DOS path variable—is neither the easiest to program nor the most convenient for users, and a much better way has been available since DOS 3.0. This better way is demonstrated by a program submitted by Martin Stitt, an independent software consultant from Oregon; he was on the development team for Software Link's PC-MOS/386 operating system. (For a review of this product plus three other multitasking operating systems, see "386 Operating Environments," Ed McNierney, January 1988, p. 60.)

1 MULTIPLE MEMORY MODEL ASSEMBLY

Although most serious applications are now written in high-level languages, developers usually have a toolbox of assembly language routines for some of the system-level or time-critical portions of the application. Ideally, the developer should be able to call these routines from any language and any memory model with no modification of the assembly source code. The current versions of Microsoft language processors—MASM 5.x, C 5.x, FORTRAN 4.x, and QuickBASIC 4.x—bring this ideal several steps closer to reality.

First of all, Microsoft has standardized the segment structure for all of the memory models in each of its programming languages; second, version 5.0 of MASM introduced several directives that automatically provide segment names and other predefined symbols that are appropriate to each memory model; and third, supplied with MASM is a file of macros (MIXED.INC) to automate some of the interfaces that vary with the memory model.

The ideal is not yet attained, however, and changing memory models still requires some changes to be made to the assembly language source code. One type of problem is that the size of the address pointers varies with the model. MASM attempts to address this by defining the assembly-time symbols **@codesize** and **@datasize**. When the assembler encounters a .MODEL directive in the source, it defines **@codesize** as 0 for the small and compact models and 1 for medium, large, and huge models. Similarly, **@datasize** is defined as 0 for the small and medium models, 1 for the compact and large models, and 2 for the huge model. In short, **@codesize** indicates whether function pointers are near or far, and **@datasize** indicates the size of data pointers.

However, the values assigned to these predefined symbols are not always useful, because they do not reflect the actual length of the pointers, which is either two or four bytes. The actual length is useful for defining structures that represent stack frames. The include file PTRLEN.INC (listing 1) solves this problem by declaring the symbols **@cplen** and **@dpen** whose values give the actual pointer lengths.

These symbols, in turn, can be used in a structure defining stack frames for all memory models, as shown in FUNC.ASM (listing 2). This code can be assembled for any model merely by changing the model designation in the .MODEL directive. Even this small change can be avoided by replacing the MODEL directive with the following sequence:

```
include MIXED.INC
ifndef model
    model equ <small >
endif
Setmodel
```

Setmodel is a macro (defined in file MIXED.INC) that issues a .MODEL directive based on the value of the symbol **model** defined with the /d switch on the MASM command line; for example:

```
masm func /dmodel = large
```

The **ifndef** statement provides a default model designation in case you forget the /d switch; otherwise, the assembly fails due to an undefined symbol in the **setmodel** macro. You could simplify the above sequence of assembly directives by inserting the **ifndef** statement into the MIXED.INC file.

MASM 5.0 has several other features that facilitate interfacing to various memory models in higher-level languages; they are fully described in chapter 5 of the MASM *Programmer's Guide*. These features make the upgrade from version 4.0 a must for programmers who value their time.

2 FINDING AUXILIARY FILES

When COMMAND.COM loads a program for execution, it can automatically search through multiple subdirectories to locate the required file, provided that the user has issued a PATH command listing the subdirectories to be searched. Most applications, however, are made up of more than one file, and the program that DOS loads to initiate the application must in turn find and load overlays or drivers. Several methods for doing this are available.

The simplest, and the least flexible method is for the program to look in only one place. This could be the current directory (Lotus 1-2-3 Release 1A

did this) or another directory whose name is hard-coded into the program. A minor improvement is to allow the user to specify this directory when the application is installed (WordPerfect, for example).

DOS 3.3 offers help for such simple-minded applications by means of the APPEND command. When DOS receives a request from a program to open a file whose name does not include a path component, DOS first searches the current directory, then the directories named in a previous APPEND command. The implementation of APPEND leaves much to be desired and robust applications should not depend on it (for details, see "The Twilight of DOS," Julie Anderson, August 1987, p. 180.)

The next step in convenience is to let the user specify the data directory in an environment variable. For example, the Brief editor looks for its help files in the directory specified as the value of the environment variable BHELP. For the application developer, this method is easy to implement (many C compilers have functions to extract the value of a specified variable from the environment); for the user, it provides a fair measure of flexibility, allowing him to easily incorporate a reshuffling of the directory structure.

The method most commonly used by well-behaved applications is to search for files in all the directories named in the DOS path. Unlike the search of the APPEND directories, which is automatically performed by DOS without any awareness on the part of the application, searching of the PATH directories must be explicitly programmed. This involves significantly more of an effort than the searching of a single directory named in an environment variable. After obtaining the path string, the program must establish a loop that parses the string, extracts an individual directory name, and attempts to open a file in that directory. A common variation is to search a series of directories named in some environment variable other than PATH; for example, the Microsoft Linker looks for library files in all the directories named in the LIB variable.

Another method, available since DOS 3.0, is possibly the most useful for the majority of applications and is the easiest to implement. That is to look for files in the same directory from which the program was loaded, called the *home directory*. This provides two advantages over the usual method of

searching along the DOS path. First, the path need not contain directories of seldom-used applications; the user can start these by explicitly specifying the directory on the command line. For example, a user who only occasionally uses QuickBASIC might not want to keep that compiler on the path, but start it from any directory by typing \BASIC\QB. However, QuickBASIC looks for its configuration file (QB.INI) in the current directory and then on the path, not in its home directory. Failing to find it, QuickBASIC comes up in default mode.

A second advantage is that an application can be kept in more than one version, with the production version on the path and an experimental or obsolete copy accessible only by explicitly naming its directory. If the auxiliary files for the various versions have the same names, the path search method cannot discriminate among them and will load the production version from the directory on the path. But if the application first looks for its files in its home directory, a nonstandard version started by explicitly naming its directory will find the correct copies of the needed files.

This method is easy to implement because, when DOS 3.x loads a program, it constructs the full path name to the program's executable file and places it in the program's environment. The environment consists of a series of ASCII strings terminated with null bytes, with the last one terminated by two 0 bytes. In all DOS versions since 3.0 (and also in PC-MOS/386), the 0 bytes are followed by a word containing a count of additional items in the environment; that value is always 1. That one item is a null-terminated string giving the full path name of the program that owns this environment.

The program can easily find this string and strip off the file name portion, giving the path to the home directory. Many C compilers simplify this even further; they produce program entry code that finds the program name string and passes a pointer to it in **argv[0]**, the first element in the array of pointers to the command-line arguments received by the program's **main** function.

For programs that do not receive this information from the entry code, Martin Stitt provides the assembly language routine PROGNAME.ASM, shown in listing 3. It is written for MASM version 4.0 and is meant to be called from the small memory model of Microsoft C; it needs a few minor modifications for other memory models, compilers, and languages.

The **progname** function receives a single argument, the address of an array to hold the program name. The function first gets the segment address of the program segment prefix (PSP) by calling interrupt 21H, function 62H. Then it reads the segment address of the environment from the word at offset 2CH of the PSP and searches the environment until it finds two adjacent bytes of zeroes. Finally the function moves the program name, which begins three bytes beyond the final zero, to the output array.

The calling program, whether it receives the program name from this function or from **argv[0]**, must trim the file and extension to obtain the path to the home directory. This is done by PATHNAME.C, shown in listing 4. It scans the input array, copies to the output array all characters from the leftmost character up to and including the rightmost backslash (\ ), and returns an integer value indicating the length of the output string. If the input string contains no path-name component, the function produces a null string and returns a value of 0.

Listing 4 also includes a main driver function that calls the two routines and prints out the full program name and the path name. In a real application, the calling routine could append the name of an auxiliary file to the path name and then attempt to open that file.

The path name obtained from the environment is reliable in all cases when DOS loads the program. For programs loaded by debuggers or executed by integrated development tools of the Quick or Turbo variety, the program name consists of only what the user typed to identify the file to be loaded. Usually this is just the file name and extension, but may also include a partial or relative path. In any case, this information can still be used by the program. If the debugger could find the file given just this information, the program can certainly find its auxiliary files on the same relative path (or in the current directory, if the path portion is null).

Unfortunately, few applications use this method for finding their auxiliary files. As shown by the procedures reproduced here, the method is quite easy to implement. It can even be used in conjunction with the prevalent method of searching on the DOS path. A program could look in its home directory first and then in each directory on the path. Try this method in the next multifile application you write; your users will thank you.

LISTING 1: PTRLEN.INC

```
;**
; PTRLEN.INC - define lengths of pointers depending on model.
;       Written by Ted Mirecki
;
; Must be preceded by .MODEL directive.  When .MODEL directive
; if used, MASM 5 automatically defines (among others) the
; following two symbols:
;    @datasize = 0 for small & medium, non-zero for other models;
;    @codesize = 0 for small & compact, 1 for other models.
;
;**

if @datasize
 @dplen equ 4
else
 @dplen equ 2
endif

if @codesize
 @cplen equ 4
else
 @cplen equ 2
endif
;** End of PTRLEN.INC
```

LISTING 2: FUNC.ASM

```
;** Example of model-independent stack frame structure
;   for ASM function called from C.
;
;   For purposes of illustration only; NOT EXECUTABLE
;
;   C function declaration:
;     func(int a, *int b, *char c, double d, *double e)
;**

        .MODEL small    ;change as appropriate
        include PTRLEN.INC      ;defines @cplen, @dplen

onstack STRUC           ;layout of stack frame
```

If Networking Is About Working Together, Why Do You Feel Like You're Doing It Alone?

You're the one responsible for networking the office.

If you've never done it before, you may feel you're on the spot. If you have, you know you are.

Which is why a lot of people in your position waste no time in getting the Businessland® Gold Plan.

The Gold Plan gives you an entire networking support organization. An account manager, systems engineers, field service technicians, trainers, and technical support professionals. Valuable people to have around.

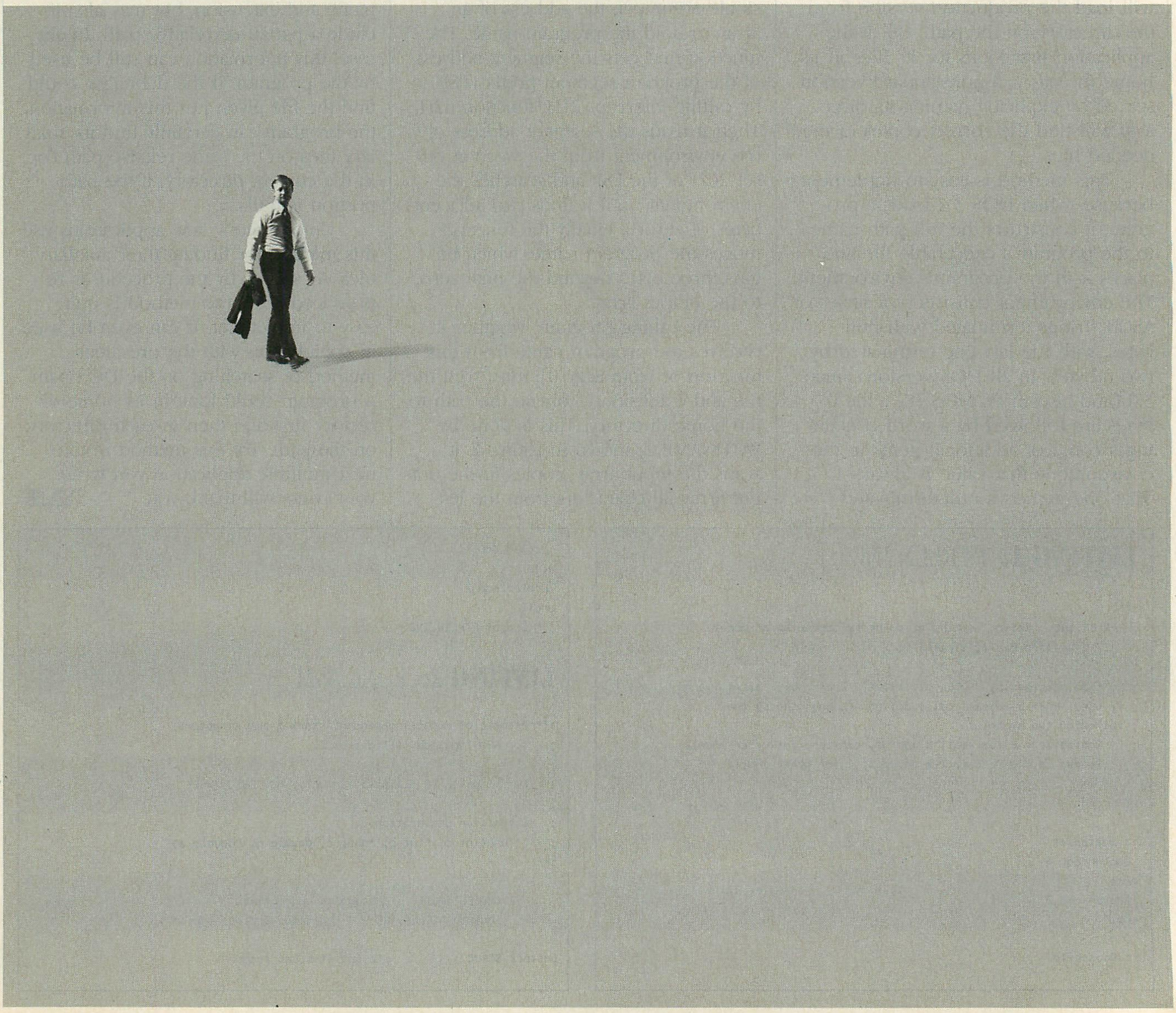
The plan includes a thorough pre-installation consultation, a detailed network plan, hardware and software installation, training for your network administrator, as well as on-going programs to serve and support you.

There's a toll-free solution line so you're never out of touch. And remote diagnostics services that let us access, and in many cases, repair your network over the phone.

If you're feeling lonely, give us a call at 1-800-328-8383 for the Businessland Center nearest you.

BUSINESSLAND®

A Different Kind of Computer Company



Businessland and the Businessland logo are registered trademarks of Businessland Inc.

CIRCLE NO. 140 ON READER SERVICE CARD

```
oldbp   dw      ?       ;BP value pushed on entry
retaddr db      @cplen dup(?)   ;return address, 2 or 4 bytes
avalu   dw      ?       ;word: value of int a
bptr    db      @dplen dup(?)   ;2 or 4 bytes: address of int b
cptr    db      @cplen dup(?)   ;2 or 4 bytes: address of char c
dvalu   dd      ?       ;double-word: value of d
eptr    db      @dplen dup(?)   ;2 or 4 bytes: address of double e
        :
        mov     AX,[BP].avalu   ;load value of parameter a
   if @datasize         ;if far data pointers...
        les     SI,dword ptr [BP].bptr  ;get 4-byte ptr into ES:SI
        mov     BX,ES:[SI]              ;load value of b
    else                        ;if near data pointers...
        mov     SI,word ptr [BP].bptr   ;get 2-byte ptr into SI
        mov     BX,[SI]                 ;load value of b
    endif
        :
        :
func    ENDP
        END             ;automatically closes segment
```

LISTING 3: PROGNAME.ASM

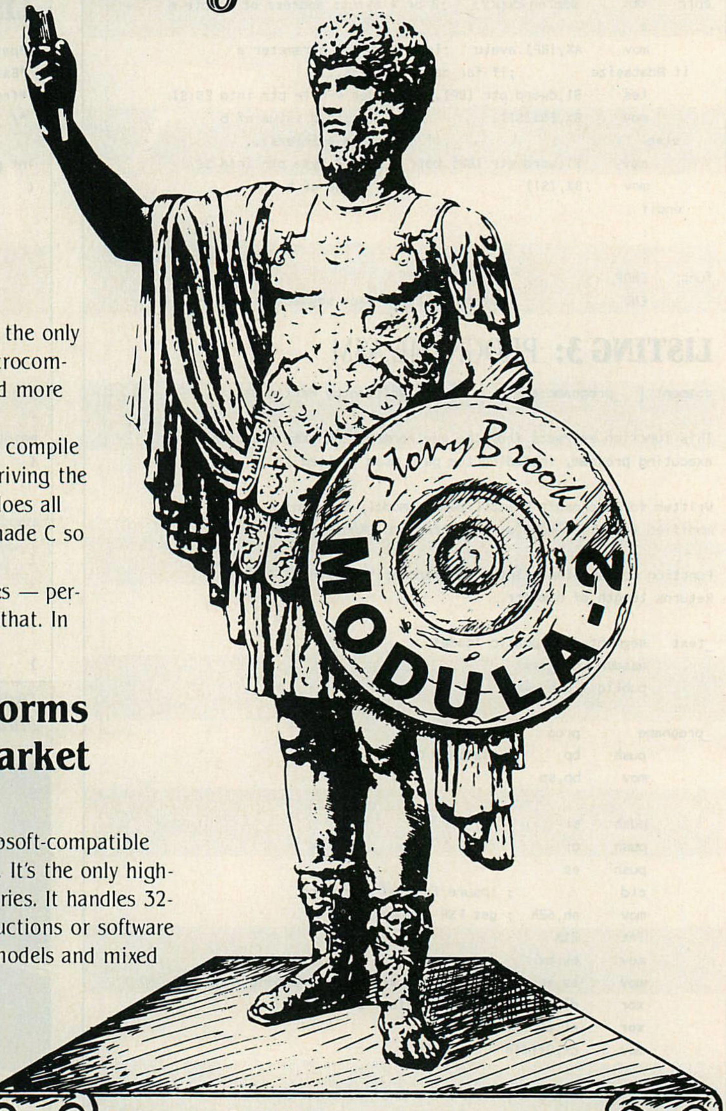
```
comment | _progname.asm                 Written by Martin Stitt

This function extracts from the environment the name of the currently
executing program, including the path used to load it.

Written for Microsoft C small memory model, but can be
modified for other models, compilers and languages.

Function declaration:  int progname(char *namestr)
Returns length of namestr.
|
_text   segment word public 'code'
        assume  cs:_text
        public  _progname

_progname       proc    near
        push    bp      ; standard C entry sequence
        mov     bp,sp

        push    si
        push    di
        push    es
        cld             ; insure forward searching
        mov     ah,62h  ; get PSP segment address
        int     21h
        mov     es,bx
        mov     es,es:[2ch]     ; get seg of environment from PSP
        xor     di,di   ; start at offset zero
        xor     al,al
        mov     cx,0ffffh

gp1:    repne   scasb   ; find the end of each env string
        cmp     byte ptr es:[di],0      ; end of environment?
        jne     gp1             ; no:  loop thru next string
        add     di,3            ; yes: point to start of name

        mov     si,di   ; save start addr of name
        repne   scasb   ; and then find its end
        mov     cx,di
        sub     cx,si   ; cx = length of string + 1
        push    cx      ; save it for return value
        mov     ax,ds   ; set up seg regs for string move
        mov     bx,es
        mov     ds,bx   ; ds:si -> start of path string
        mov     es,ax   ; es:di -> caller's receiving addr
        mov     di,[bp+4]
        rep     movsb   ; move the path string, incl. null
        mov     ds,ax   ; restore data segment

        pop     ax      ; get length of string
        dec     ax      ; adjust for terminator

        pop     es
        pop     di
        pop     si
        pop     bp
        ret
;       ret     2       ; use this if your caller does not
                        ; remove parms from stack
_progname       endp
_text   ends
        end
```

LISTING 4: PATHNAME.C

```
/*pathname function                         Written by Ted Mirecki
 *Extracts the path portion, up to & including the rightmost backslash,
 *from a fully qualified filename. Returns length of pathname string.
 */

int pathname( char *fname, char *path)
{
        int x, len;

        len = strlen(fname);
        while (len && fname[len-1] != '\\') --len;
        for (x=0; x<len; x++)
                path[x] = fname[x];
        path[x] = 0;
        return (x);
}

main()          /* demo driver for pathname & progname functions*/
{
        int  len1, len2;
        char filestr[80];
        char pathstr[80];

        len1 = progname(filestr);
        len2 = pathname(filestr, pathstr);
        printf("Full name is %s, length %d\n", filestr, len1);
        printf("Path name is %s, length %d\n", pathstr, len2);
}
```

Listings can be downloaded using PCTECHline, 301/740-8383. Parameters: 2400/1200/300 bps, no parity, 8 data bits, 1 stop bit.

I have come to bury C, sir, not to praise it.



C served us well in the days of kilobytes and kilohertz. It was the only language we could implement efficiently on our newborn microcomputers. But with today's mega-machines, shouldn't we demand more from our compilers?

Modula-2 increases productivity by catching your errors at compile time. You'll easily modularize and structure your programs, driving the hordes of barbaric bugs into the hinterlands. And Modula-2 does all this without taking away the low-level machine access that made C so popular.

Until now, you had to pay a price for the Modula-2 advantages — performance just didn't measure up to C. But we've changed all that. In a suite of benchmarks developed by PC Week:

Stony Brook Modula-2 outperforms the best C compilers on the market

(and no other Modula-2 compiler even comes close).

Stony Brook Modula-2, for 80x86 machines, produces Microsoft-compatible objects, and fully supports both Microsoft Windows and OS/2. It's the only high-level language compiler that lets you write dynamic link libraries. It handles 32- and 64-bit real numbers with in-line 80x87 coprocessor instructions or software emulation. And Stony Brook Modula-2 supports six memory models and mixed model programming.

You might want to bury your C compiler once you have used Stony Brook Modula-2, but you won't have to. We made it possible to directly call C and other languages from Modula-2, so you won't have to throw away your investment in C code.

So, friends, programmers, and C-users, lend us your ears. Call us or write for more information, and to find out how you can get a demo compiler.

Benchmark	Stony Brook M-2 V1	Logitech M-2 V3	Turbo C V1	Microsoft C V4
Magnifier	0.05	0.06	0.05	0.05
For Loop	0.28	0.73	0.44	0.68
While Loop	0.35	0.75	0.49	0.66
Repeat Loop	0.33	0.68	0.38	0.66
Literal Assign	0.33	0.97	0.60	0.94
Memory Access	0.33	1.04	0.55	1.00
Real Arithmetic	4.68	13.40	3.73	2.96
Real Algebra	4.18	12.87	3.07	3.64
Vector	0.77	1.75	1.21	1.26
Equal If	0.66	1.47	0.93	1.32
Unequal If	0.66	1.43	0.99	1.32
No Parameters	0.33	0.60	0.34	1.04
Value	0.50	0.71	0.60	1.21
Reference	0.61	0.77	0.66	1.21
Real Maths	3.90	4.61	7.85	18.51

The compiler package includes DOS runtime library objects and full sources for our split-screen text editor for $195. The development package includes all of the above plus an automatic make utility, a symbolic debugger, and run-time library sources for $345. MasterCard and Visa accepted. Add $5 for shipping in North America, $25 for overseas shipping.

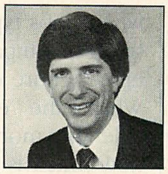
StonyBrook INC. SOFTWARE

Forest Road, Wilton, New Hampshire 03086 ● (603)654-2525 Ask for Cleopatra.

CIRCLE NO. 159 ON READER SERVICE CARD

OUTFITTING THE END USER

Surviving Your Success



P.C.Coffee

PC wizards must speak the language of costs and benefits to advise management about their company's computing needs.

The funny thing about this business is that even success may bring nasty moments of tension. Suppose you get your fondest wish: the president of the company becomes a full-fledged power user, firmly convinced that every desk should have a workstation—telecommuting, on-line filing systems, the whole ball of wax.

Then the president tells you, in your capacity as corporate systems integrator, "We have to make a choice: we can lease enough systems to equip the whole company right away, or we can buy them over the next four years. I'd like your recommendation by the end of the week." What do you say?

Tough choice, isn't it? If you lease for the entire company in one swoop, you get the benefits of PC pervasion. The old paper-based systems can die a painless death if you can bring everyone into the 1980s at once. The cost-effectiveness of networking is greater and the benefits more obvious if a critical mass can be achieved.

The dark side of this tidal-wave scenario is an incredible surge of demand for system configuration, user support, software recommendations, and other essential logistics. You don't want the president stumbling over expensive machines stacked in the halls awaiting delivery and setup.

As you mull over your decision, that little voice in your head might say, "But the lease cost is based on a lifetime of three years. Even our 5-year-old PCs are as useful today as when we bought them; our standards have merely risen." Incremental buying may be preferable to "total immersion" leasing if aging machines can be trickled down as smart terminals and text-editing stations, with the most compute-intensive tasks always getting the benefit of the most current hardware.

Either approach has factors in its favor. Your credibility depends on making a well-founded recommendation, rooted in the business requirements of the organization, based on your presumably superior understanding of the technology—both what it can do and where it is going. Foreign as the idea may be, the time has come to quantify the costs and benefits.

COUNTING UP THE BEANS

Costs and benefits, benefits and costs. A preoccupation with short-run financial measures of success is blamed for some of the economy's most fundamental problems, but this does not eliminate the need for a command of "the financials" among those who would change the entire character of business computing. Building yourself a niche as the corporate "PC wizard" may be fun for awhile, but wizards don't make policy; they just get stuck with making the policies work.

The better strategy is to have your input earlier in the game. This may mean learning to speak a new language—just in time to use it for delivering bad news. There are some techniques for laying out the facts so that the surprises are less unpleasant.

Experienced mainframe managers are used to defending the return on their substantial capital assets. A company that puts $3 million and change into a box with lights and knobs on the front expects to be told, in no uncertain terms, what it is getting in return and why the same job can't be done for half the price. In the early years of personal computing, however, the gains were obvious and huge while the costs could be buried elsewhere in the budget. In such an environment, you could successfully evade the notion of planning; it was clearly more profitable to go ahead and do it. This is less and less the case.

Let me stress one point from the outset: I am not claiming that you can rigorously predict the costs or benefits that will flow from specific actions, let alone make the measurements needed to prove that you were right at year's end. The work that gets supported by personal computing usually does not lend itself to this kind of analysis; indeed, many of the improvements are in quality rather than quantity.

My purpose instead is to furnish insight into the relative importance of various factors. How much more powerful does a system have to be, for example, to justify six months rather than a week to master its capabilities? Is the answer different in an environment

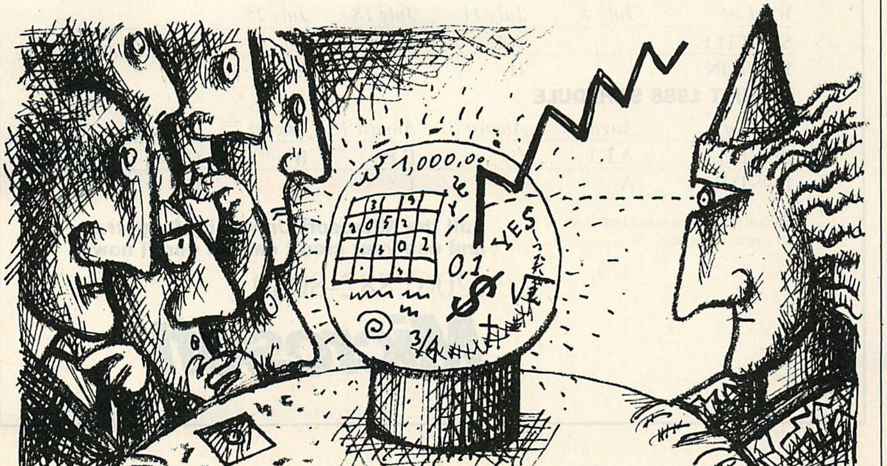


ILLUSTRATION • MACIEK ALBRECHT

where jobs typically turn over in a year or less, compared to those with an average tenure of three years or more? What if the "easy-to-use" system is considerably more expensive? Conversely, how much is it worth to develop job-specific software for the "difficult" system, either on contract or in-house?

One of the problems of assessing costs and benefits is a natural consequence of the ubiquitous bell-shaped curve—the one that your high school teacher drew to grade exams. Every staff has varying levels of motivation, from gung-ho to premature retirement. This natural distribution interacts with PCs and budgets to produce what I call the Slippage Syndrome.

In the early years of the PC, obtaining a computer for your department, let alone your individual desk, was difficult. Even so, enough of these personal systems were out there that the term *power user* became established virtually overnight. We may reasonably assume that the majority of those personal systems were on the desks of people who were already working harder and smarter, the kind who actively seek out better ways to do more and better work—in other words, people who stand at the top of the bell curve of performance.

So far, so good: the tools were going where they could best be used. The problem arises when the people responsible for measuring costs and benefits take the output "After Computer" (A.C.) and subtract the output "Before Computer" (B.C.) to figure out how much the computer is contributing. Somebody, please, teach these people to divide: the *ratio* of improvement is what matters. Computers do not add to your ability; all they can do is multiply it, and even a factor of ten times zero will still be zero.

But the trap is set. The costs of the first wave of machines are subtracted from their estimated benefit, and the net value is used to justify the next wave. But the new machines usually go onto the desks of the next most capable and/or next most motivated segment of the work force.

How much of a problem could this possibly be? If the members of the second group were, on the average, three-fourths as productive B.C. as the early adopters who led the PC charge, and if the first wave of machines paid for themselves twice over in the first year, then the second-wave machines will pay for themselves only 1.5 times over. Not bad at all, but it means that the *net* benefit is only half of what was expected by management:

$$\frac{(1.5 - 1)}{(2 - 1)} = .50$$

The numbers get much worse. By the time you get all the way down to the lower tail of the bell curve, the same *ratio* of personal performance improvement may put you merely at the break-even point after you allow for the reduced value of benefits that lie farther into the future—the so-called "present value." Every successive measurement of the return on investment from additional computer purchases shows a smaller net benefit than the one before it.

Management suddenly sees that the total investment in micro hardware and software is beginning to rival the comparable figure for the mainframes. Someone notices that $300 software packages are being controlled more like stationery than like the expensive products they are. (Do you wonder

how many of the copies of Lotus 1-2-3 that your company bought are still on the premises?) Users who are not prepared to learn macro editing on their own time start to ask for training. These are costs that may never have appeared as a factor in the early years.

Crisis is in the air. Some managements—maybe yours—will panic. Firms that are big enough to know better have frozen computer purchases for months at a time. The problem is not that the systems have ceased to be a good investment: they have just ceased to be unbelievably good and have become merely excellent.

HIGH-PRICED SPREADS

You can avoid the pitfall of constantly painful surprise by preparing management for the life cycle ahead. First and foremost, the forecasts made at the head of the energy curve—the energy of the users, that is—must not be extrapolated to the tail. Look at the real costs of equipping your users and the benefits that can be reasonably expected as separate components. Both will vary with the application, the physical environment, and the makeup of the work force on a department-by-department basis. Take the net benefit only at the end of the calculation; never use the net benefit as a figure with a life of its own.

How do you model the cost and benefit components? You begin with time. Designate a row across the top of a spreadsheet to represent years, from 0 (the present) up to as many as you plan to look ahead.

The next step is to model your technology strategy. You can predict future system purchases in either of two ways: I call them TOP (threshold of pain) and MUP (maximum usable performance). The TOP model assumes that systems wind up costing as much as can be approved by a reasonably low level of management, say $5,000. This was enough to buy a PC in 1982, an AT in 1985, and an 80386-based machine or Mac II, in 1987.

The basic TOP assumption—the cost of new systems will remain roughly constant while performance rises steadily—was a good one for almost all users until quite recently. This was simply because even a moderate-intensity user could bring a $5,000 system to its knees. The demand for higher performance was widespread.

Today, though, 10- and 12-MHz 80286-based machines are a commodity product; many users are literally unable to use any more performance than this class of machine provides (how many times a day do you do a 20-page global search and replace?). Your future purchases, therefore, may trend toward MUP behavior, where cost declines fairly rapidly while performance rises only slowly.

Either way, you can enter the technology forecast into the spreadsheet as a row of values representing the average cost per system predicted in each of the coming years. With a TOP model, this is easy: just put the value you are expecting in each column, as far across as you want to make the model go. The MUP scenario is a little more complicated.

Although MUP cost trends can be easily modeled as a constant fractional drop each year—for example, each year's cost being 85 percent of the cost in the prior year—this may not be completely accurate. I doubt that you will ever be able to buy an AT clone for one-tenth of its 1988 cost, even in

the year 2002 (14 years at 15 percent decline per year, compounded, equals an 89.7 percent decline). At some point, costs stabilize as a technology matures; power supplies, keyboards, and video displays just aren't becoming cheaper very quickly.

If you agree with the idea of a "floor" that is reasonable to regard as absolute, then you can model cost versus time as follows. If today's cost is C, and the floor value is F ($0 < F < C$), then you can predict the cost in year T (starting from $T = 0$) as

$$F + (C - F) * exp\left(\frac{-5T}{t_{99F}}\right)$$

where t_{99F} is the year by which you expect the cost to be about 99 percent of the way to its floor. The **exp** function is provided by most spreadsheets, or you can use 2.7 to the power of the argument as an approximation.

A quick math check: if you model today's price as $2,600, the floor price as $1,200, and use five years as the value of t_{99F}, then the price-per-system forecast in year 1 should be $1,715; in year 5, it should be $1,209.

In making the choice between TOP and MUP, you have to answer some fundamental questions about future system purchases. The choice need not be strictly one or the other: you might envision your organization buying on a TOP model for the next several years, then transitioning to MUP as your needs are finally exceeded. If so, then model it that way. The point is to get a model that you are prepared to defend as representative of what you expect to happen.

What about the other component of costs—that is, the continuing costs of ownership, including new software and/or software upgrades as well as hardware upgrades and maintenance. These costs must be factored in for a meaningful forecast. I once ran a fairy-tale scenario in which several hundred additional systems were *given* to a company in year 0; the effect on total capability six years later was negligible unless the budget for upgrades and maintenance was increased in proportion. The newer systems simply swamped the contribution of the surviving old ones when the effect of technology improvement was compounded by the effect of mechanical attrition. Disks and printers are like light bulbs: the question is not whether they will break, but when.

I suggest a low figure for continuing costs during the first year—say, 5 percent of the initial system cost. This reflects the assumption that your initial software will probably be adequate while the hardware will mostly be under warranty. The costs will probably stay low through the second or third year, then skyrocket in the third or fourth year to as much as 35 percent of the initial system cost as upgrades (storage, processor, and output devices) are made. Beyond this time, the fraction will probably drop off to a maintenance level for the remainder of the system's useful life.

Going back to your spreadsheet, you may want to show number of systems per year in the next row, below the forecast cost per system, with the cost of the new systems to be purchased in each year below that. The continuing costs of each year's cohort of systems can be a set of separate rows, each beginning in the year of purchase and continuing as long as you care to predict. This makes it easy to visually associate costs with purchases.

REAP THE WHIRLWIND

Now that you have modeled the key components of cost, you can turn your attention to benefits. The important point here is to capture the delay between initial purchase and real returns. You have already used an exponential function to model a value that approaches some constant level; you can apply it to PC productivity as well.

Assume that any new system returns zero benefit during initial familiarization; further assume that the variables are (1) the level of benefit ultimately produced and (2) how long it takes to reach this level. If the steady-state benefit is B (in cost reduction and/or output quality/quantity improvement per year), and the time to reach 99 percent of this level is t_{99B}, then the level of benefits in year T will be

$$B * \left[1 - exp\left(\frac{-5T}{t_{99B}}\right)\right]$$

Math check: if B is $2,000/year, and t_{99B} is 5 years, then the value of benefits in year 1 should be $1,264; in year 5, $1,987. Benefits, for each year's cohort of systems, become another row in your spreadsheet.

You may not like using years as the time interval for your benefit calculations, because it ignores the difference between a system that users master in a week compared to a system that takes several months. If you prefer, use months (or even weeks) instead, but make sure that you adjust the val-

ues of B and T, as well as the value of t_{99B} to reflect this change in units.

All these calculations may seem hopelessly complicated, but look at what you now have. You can reduce arguments over future PC/workstation technology to three issues: how much you will pay for your systems (a function of C, F, and t_{99F}); how much they will do once users understand them (B); and how long you will keep them (the number of columns that you carry your forecast across).

You can reduce arguments over ease of use to one issue: how long it takes the typical user to reach a comfortable level of proficiency (t_{99B}). You can evaluate training alternatives on debatable but still concrete factors: how much it will cost and how it will affect the values of B (more skills) and t_{99B} (faster learning).

The final step is to take the differences between total cost and total benefit, year by year. Bring the net for each column to the bottom line. Now you can use the Net Present Value function in your spreadsheet to figure out what various strategies are worth compared to cash in hand today.

You can also compare the strategy of outfitting an entire department at the same time (greater benefits due to pervasion, but each department has to learn from scratch) with the alternative of introducing PCs gradually throughout the company (more gradual buildup of benefits, but experienced users will be able to reduce the learning time in the outyears).

Some people argue that a cost/benefit analysis for computers is a waste of effort. Their argument goes like this: you don't do a cost/benefit analysis to decide whether or not to fix a leaky roof, because the repair is just a basic factor in providing a well-equipped workplace; computing today is just as fundamental. I agree with the sentiment, but it omits an important point. The model that we have developed here helps you *communicate* with people whose job is deciding between many equally attractive opportunities for deployment of limited corporate assets. Speaking this language may make the difference in putting your ideas high on management's list of strategic opportunities.

Peter C. Coffee is managing partner of SolveWare, a developer and business computing consultant, and is active in AI and distributed computing applications for aerospace and educational clients.

INDEX TO PRODUCTS

PC TECH JOURNAL

DIRECT ORDER EXPRESS

Instant access to the latest products and services is just a phone call away

Call or write for the latest catalog

DISCOVER PARADISE

Programmer's Paradise Gives You Superb Selection, Personal Service and Unbeatable Prices!

Welcome to Paradise. The microcomputer software source that caters to your programming needs.
Discover the Many Advantages of Paradise...

- Lowest price guaranteed
- Latest versions
- Huge inventory, immediate shipment
- Knowledgeable sales staff
- Special orders
- 30-day money-back guarantee

Over 500 brand-name products in stock—if you don't see it, call!

We'll Match Any Nationally Advertised Price.

	LIST	OURS
386 SOFTWARE		
386-TO-THE-MAX	75	66
ADVANTAGE 386 C OR PASCAL	895	839
DESQVIEW	130	115
FOXBASE+/386	595	459
HIGH C-386	895	839
MICROPORT SYS V/386 (COMPLETE)	799	679
MS WINDOWS/386	195	130
NDP C OR FORTRAN-386	595	553
PHARLAP 386|ASM/LINK	495	422
SCO XENIX SYS V 386 (COMPLETE)	1495	1199
VM/386	245	182
X-AM	595	549
ARTIFICIAL INTELLIGENCE		
ARITY STANDARD PROLOG	95	80
MULISP-87 INTERPRETER	300	199
PC SCHEME	95	86
TURBO PROLOG	100	69
TURBO PROLOG TOOLBOX	100	69
ASSEMBLERS/LINKERS		
ADVANTAGE DISASSEMBLER	295	279
MS MACRO ASM (DOS OR OS/2)	150	99
OPTASM	195	172
PASM86	195	115
PLINK86PLUS	495	279
BASIC		
DB/LIB	139	121
FINALLY!	99	90
FLASH-UP	89	80
MACH 2	75	66
MS BASIC COMP. 6.0 (DOS OR OS/2)	295	189
MS QUICKBASIC	99	69
QUICKPAK	69	60
QUICKWINDOWS W/SOURCE	99	90
TRUE BASIC	100	80
TURBO BASIC	100	69
TURBO BASIC TOOLBOXES	100	69

DBASE TOOLS

APPLICATION PLUS	499	279
DBASE III PLUS	695	399
DBASE TOOLS FOR C OR PASCAL	90	69
DBFAST	69	60
DEBUG III	195	181
FOX TOOL BOX	295	267
FRIENDLY FINDER	99	90
GENIFER	395	282
HI-SCREEN XL	149	129
QUICK ENTRY	99	90
R&R	150	139
REPORT PLUS	150	131
THE DOCUMENTOR	295	249
TOM RETTIG'S LIBRARY	100	80
UI PROGRAMMER	295	249

C COMPILERS		
LATTICE C	500	272
MICROSOFT C (DOS OR OS/2)	450	285
QUICK C	99	69
TURBO C	100	69
C INTERPRETERS		
C-TERP	298	232
INSTANT C	495	384
RUN/C	120	85
RUN/C PROFESSIONAL	250	159

	LIST	OURS
C LIBRARIES		
C ASYNCH MANAGER	175	137
C-FOOD SMORGASBORD	150	97
C TOOLS PLUS/5.0	129	101
C UTILITY LIBRARY	185	125
ESSENTIAL COMMUNICATIONS	185	125
COMMUNICATIONS PLUS	250	199
GREENLEAF C SAMPLER	95	69
GREENLEAF COMM LIBRARY	185	125
GREENLEAF FUNCTIONS	185	125
MULTI-C	149	137
PFORCE	395	215
RESIDENT C W/SOURCE	198	169
TIMESLICER	295	279
TURBO C TOOLS	129	101
COBOL		
E-Z PAGE	295	269
MICRO FOCUS		
COBOL/2	900	733
COBOL/2 TLSET	900	733
PC-CICS	1500	1189
LEVEL II COBOL	349	282
PERSONAL COBOL	149	119
MICROSOFT COBOL	700	452
MICROSOFT SORT	195	130
OPT-TECH SORT	149	105
REALCICS	995	799
REALIA COBOL	995	794
W/REALMENU	1145	899
RM/COBOL	950	763
RM/COBOL-85	1250	999
RM/SCREENS	395	339
SCREENIO	400	382
COMMUNICATIONS		
ASCOM IV	195	177
CARBON COPY PLUS	195	142
CO-SESSION (2 USER)	249	227
SUPPORT	175	157
APPLICATION	125	116
PTEL	50	45
SIDETALK	120	90
DEBUGGERS		
ADVANCED TRACE-86	175	121
PERISCOPE I	345	282
PERISCOPE II	175	141
PERISCOPE III 8 MHZ	1095	899
PERISCOPE III 10 MHZ	1195	979
PFIX 86 PLUS	395	215
DISK/DOS/KEYBOARD UTILITIES		
ADVANCED NORTON UTILITIES	150	101
COMMAND PLUS V. 2.0	80	70
DISK OPTIMIZER	75	56
FETCH	55	49
NORTON COMMANDER	75	56
PC TOOLS DELUXE	80	70
PDISK	145	107
VFEATURE	80	75
EDITORS		
BRIEF	195	CALL
W/DBRIEF	275	CALL
EMACS	295	268
EPSILON	195	151
KEDIT	125	101
MKS Vi	75	66
MULTI-EDIT	99	90
NORTON EDITOR	75	70
PC/EDT+	295	269

	LIST	OURS
PMATE	195	115
SPF/PC	245	185
VEDIT PLUS	185	131
FILE MANAGEMENT		
BTRIEVE	245	185
XTRIEVE	245	189
REPORT OPTION	145	109
BTRIEVE/N	595	455
XTRIEVE/N	595	459
REPORT OPTION/N	345	279
CBTREE	159	141
C-TREE	395	318
R-TREE	295	241
C-TREE/R-TREE BUNDLE	650	523
D-TREE	395	CALL
DBC III	250	172
DBC III PLUS	750	599
DB_VISTA OR DB_QUERY	195	CALL
SINGLE USER W/SOURCE	495	CALL
MULTIUSER	495	CALL
MULTIUSER W/SOURCE	990	CALL
INFORMIX PRODUCTS	CALL	CALL
XQL	795	599
FORTRAN COMPILERS		
LAHEY FORTRAN F77L-EM/16	695	629
LAHEY PERSONAL FORTRAN 77	95	86
MS FORTRAN (DOS OR OS/2)	450	285
RM/FORTRAN	595	479
FORTRAN LIBRARIES/UTILITIES		
DIAGRAM'ER OR DOCUMENT'ER	129	115
GRAFMATIC OR PLOTMATIC	135	119
MAGUS NUMERICAL ANALYST	295	252
MATHPAC	495	445
SPINDRIFT LIBRARY	149	135
SSP/PC	350	272
GRAPHICS		
ADVANTAGE GRAPHICS (C)	250	229
ESSENTIAL GRAPHICS	299	229
W/SOURCE	598	509
GSS GRAPHIC DEV. TOOLKIT	495	399
HALO '88	325	229
HALO '88 (5 MICROSOFT LANG.)	595	399
METAWINDOW PLUS	275	232
TURBOWINDOW/C	95	80
TURBO HALO (FOR TURBO C)	99	80
MODULA-2		
LOGITECH MODULA-2		
COMPILER PACK	99	81
DEVELOPMENT SYSTEM	249	199
TOOLKIT	169	141
SOLID B+ TOOLBOX	99	89
STONYBROOK MODULA-2	195	179
W/UTILITIES	345	299
OBJECT-ORIENTED PROGRAMMING		
ACTOR	495	423
ADVANTAGE C++	495	479
PFORCE++	395	215
SMALLTALK/V	100	85
APPLICATION PACKS	50	45
SMALLTALK/V286	200	169
OPERATING SYSTEMS		
MICROPORT DOS MERGE	149	129
MICROPORT SYS V/AT	549	469
SCO XENIX SYSTEM V (COMP.)	1295	999
WENDIN-DOS	99	80
OTHER MICROPORT, SCO, WENDIN PRODUCTS	CALL	CALL

	LIST	OURS
PASCAL COMPILERS		
MICROSOFT PASCAL(DOS OR OS/2)	300	189
PASCAL-2	229	CALL
TURBO PASCAL	100	69
TURBO PASCAL DEV. LIB.	395	289
BORLAND ADD-ONS	CALL	CALL

FEATURED PRODUCTS

ESSENTIAL COMM LIBRARY 2.0—New version includes driven support for 8 ports, baud rates up to 38,400 bps., transmit and receive buffering, a complete set of Hayes compatible modem functions and more.
List: $185 **Ours: $125**

w/ Breakout, the async comm debugger
List: $125 **Ours: $89**

GREENLEAF C SAMPLER—For the Quick C or Turbo C programmer. A selection of over 100 of the best functions from Greenleaf's other popular products, The Greenleaf Functions, The Greenleaf Comm Library, and Greenleaf Data Windows. Includes logical windows, pull-down menus, intelligent keyboard and time and date functions, interrupt communications and much more.
List: $95 **Ours: $69**

SMALLTALK V/286—Latest version of Smalltalk/V takes advantage of the power of 286 and 386 systems, running in protected mode and addressing up to 16 meg directly. Smalltalk V/286 operates at least twice as fast as Smalltalk/V, and includes support for multitasking.
List: $200 **Ours: $169**

SORTEX—Flexible, general purpose sort/merge facility. No practical limitations on the file size, record size, field size, number of records or number of keys. Comprehensive error reporting.
List: $99 **Ours: $89**

TURBO PASCAL ADD-ONS		
ASCII TURBO GHOST WRITER		
STARTER	99	80
COMPLETE	289	262
DOS/BIOS & MOUSE TOOLS	75	70
FLASH-UP	89	80
METRABYTE DATA ACQ. TOOLS	100	90
SCREEN SCULPTOR	125	96
SYSTEM BUILDER	150	131
IMPEX	100	90
REPORT BUILDER	130	116
T-DEBUG PLUS V. 4.0	45	39
W/SOURCE	90	80
TURBO.ASM	99	70
TURBO ASYNCH PLUS	129	101
TURBO GEOMETRY LIBRARY	100	90
TURBO HALO	99	80
TURBO MAGIC	99	90
TURBO POWER TOOLS PLUS	129	101
TURBO POWER UTILITIES	95	79
TURBO PROFESSIONAL 4.0	99	80
TURBO WINDOW/PASCAL	95	80
SCREENS/WINDOWS		
C-SCAPE	299	282
CURSES W/SOURCE	250	172
GREENLEAF DATA WINDOWS	295	209
HI-SCREEN XL	149	129
JYACC FORMAKER	495	453
JYACC JAM	750	684
MICROSOFT WINDOWS	99	69
MS WINDOWS DEVELOPMENT KIT	500	319
PANEL PLUS	495	395
PANEL/QC OR /TC	129	99
SCREENSTAR W/SOURCE	198	169
VIEW MANAGER	275	199
VITAMIN C	225	162
VC SCREEN	99	80
WINDOWS FOR DATA	295	239
W/SOURCE	590	479

XENIX/UNIX SOFTWARE

MICROPORT & SCO PRODUCTS	CALL	CALL
ADVANTAGE C++	695	625
BTRIEVE/N	595	455
DIRECTORY SHELL (286)	349	315
DIRECTORY SHELL (386)	495	445
EPSILON	195	152
FOXBASE+/286	795	599
INFORMIX PRODUCTS	CALL	CALL
JYACC FORMAKER	895	809
JYACC JAM	1350	1224
KORN SHELL	145	115
MICROSOFT LANGUAGES	CALL	CALL
PANEL PLUS	795	675
RM/COBOL	1250	954
RM/FORTRAN	750	553
WINDOWS FOR DATA	795	CALL

ADDITIONAL PRODUCTS		
ADVANTAGE VCMS	379	339
BABY/36 (RPG II) *NEW*	3000	2699
BASTOC	495	399
DAN BRICKLIN'S DEMO PROGRAM	75	59
DEMO PROGRAM II	195	179
DB2C	299	272
FLOW CHARTING II	229	205
MAGIC PC	195	179
MKS RCS	189	162
MKS-SQPS	495	473
MKS TOOLKIT	169	149
MS OS/2 PROG. TOOLKIT *NEW*	350	229
NORTON GUIDES	100	65
PC-LINT	139	101
POLYMAKE	149	131
POLYTRON PVCS	CALL	CALL
PRE-C	295	159
SOURCE PRINT	95	81
TREE DIAGRAMMER	77	70

Terms and Policies
- **We honor MC, VISA, AMERICAN EXPRESS** No surcharge on credit card or C.O.D. Prepayment by check. New York State residents add applicable sales tax. Shipping and handling $3.95 per item, sent UPS ground. Rush service available, prevailing rates.
- Programmer's Paradise will match any current nationally advertised price for the products listed in this ad.
- Prices and Policies subject to change without notice.
- Hours 9AM EST—7PM EST
- We'll Match any Nationally Advertised Price
- ***Mail Orders include your phone number***

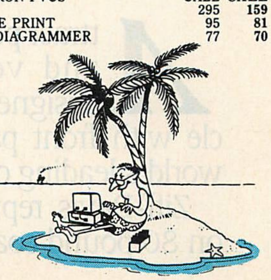
*Ask for details. Some manufacturers will not allow returns once disk seals are broken.

Dealers and Corporate Buyers—Call for special discounts and benefits!

1-800-445-7899
In NY: 914-332-4548
Customer Service: 914-332-0869
International Orders: 914-332-4548
Telex: 510-601-7602

CIRCLE NO. 173 ON READER SERVICE CARD

Programmer's Paradise™



A Division of Hudson Technologies, Inc.
42 River Street, Tarrytown, NY 10591

Seconds Never Looked So Good!

Attract potential customers, distributors, and vendors with our tastefully designed reprints. Revitalize your article with front page coverage from one of the world's leading computer magazines.

Ziff-Davis reprints are concisely reformatted on 80 pound coated paper and printed from full color to black and white.*

To find out how you can have your article reprinted, call or write today.

Jennifer Locke-Reprints Manager, Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016, (212) 503-5447.

*minimum order 500 reprints.

UNLEASH YOUR 80386!

Your 80386-based PC should run two to three times as fast as your old AT. This speed-up is primarily due to the doubling of the clock speed from 8 to 16 MHz. The new MicroWay products discussed below take advantage of the real power of your 80386, which is actually 4 to 16 times that of the old AT! These new products take advantage of the 32 bit registers and data bus of the 80386 and the Weitek 1167 numeric coprocessor chip set. They include a family of MicroWay 80386 compilers that run in protected mode and numeric coprocessor cards that utilize the Weitek technology.

The benefits of our new technologies include:

• An increase in addressable memory from 640K to 4 gigabytes using MS-DOS or Unix.

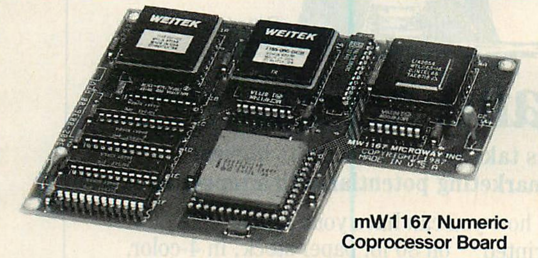
• A 12 fold increase in the speed of 32 bit integer arithmetic.

• A 4 to 16 fold increase in floating point speed over the 80387/80287 numeric coprocessors.

Equally important, whichever MicroWay product you choose, you can be assured of the same excellent pre- and post-sales support that has made MicroWay the world leader in PC numerics and high performance PC upgrades. For more information, please call the Technical Support Department at

617-746-7341

After July 1988 call 508-746-7341



mW1167 Numeric Coprocessor Board

MicroWay® 80386 Support

MicroWay 80386 Compilers

NDP Fortran-386 and **NDP C-386** are globally optimizing 80386 native code compilers that support a number of Numeric Data Processors, including the 80287, 80387 and mW1167. They generate mainframe quality optimized code and are syntactically and operationally compatible to the Berkeley 4.2 Unix f77 and PCC compilers. MS-DOS specific extensions have been added where necessary to make it easy to port programs written with Microsoft C or Fortran and R/M Fortran.

The compilers are presently available in two formats: Microport Unix 5.3 or MS-DOS as extended by the Phar Lap Tools. MicroWay will port them to other 80386 operating systems such as OS/2 as the need arises and as 80386 versions become available.

The key to addressing more than 640 kbytes is the use of 32-bit integers to address arrays. NDP Fortran-386 generates 32-bit code which executes 3 to 8 times faster than the current generation of 16-bit compilers. There are three elements each of which contributes a factor of 2 to this speed increase: very efficient use of 80386 registers to store 32-bit entities, the use of inline 32-bit arithmetic instead of library calls, and a doubling in the effective utilization of the system data bus.

An example of the benefit of excellent code is a 32-bit matrix multiply. In this benchmark an NDP Fortran-386 program is run against the same program compiled with a 16-bit Fortran. Both programs were run on the same 80386 system. However, the 32-bit code ran 7.5 times faster than the 16-bit code, and 58.5 times faster than the 16-bit code executing on an IBM PC.

NDP FORTRAN-386™ $595
NDP C-386™ $595

MicroWay Numerics

The **mW1167™** is a MicroWay designed high speed numeric coprocessor that works with the 80386. It plugs into a 121 pin "Weitek" socket that is actually a super set of the 80387. This socket is available on a number of motherboards and accelerators including the AT&T 6386, Tandy 4000, Compaq 386/20, Hewlett Packard RS/20 and MicroWay Number Smasher 386. It combines the 64-bit Weitek 1163/64 floating point multiplier/adder with a Weitek/Intel designed "glue chip". The mW1167™ runs at 3.6 MegaWhetstones (compiled with NDP Fortran-386) which is a factor of 16 faster than an AT and 2 to 4 times faster than an 80387.

mW1167 16 MHz $1495
mW1167 20 MHz $1995

Monoputer™ – The INMOS T800-20 Transputer is a 32-bit computer on a chip that features a built-in floating point coprocessor. The T800 can be used to build arbitrarily large parallel processing machines. The Monoputer comes with either the 20 MHz T800 or the T414 (a T800 without the NDP) and includes 2 megabytes of processor memory. Transputer language support from MicroWay includes Occam, C, Fortran, Pascal and Prolog.

Monoputer T414-20 with 2 meg[1] ... $1495
Monoputer T800-20 with 2 meg[1] ... $1995

Quadputer™ can be purchased with 2, 3 or 4 transputers each of which has 1 or 4 megabytes of memory. Quadputers can be cabled together to build arbitrarily fast parallel processing systems that are as fast or faster than today's mainframes. A single T800 is as fast as an 80386/mW1167 combination!

Biputer™ T800/T414 with 2 meg[1] $3495
Quadputer 4 T414-20 with 4 meg[1] ... $6000

[1] Includes Occam

80386 Multi-User Solutions

AT8™ – This intelligent serial controller series is designed to handle 4 to 16 users in a Xenix or Unix environment with as little as 3% degradation in speed. It has been tested and approved by Compaq, Intel, NCR, Zenith, and the Department of Defense for use in high performance 80286 and 80386 Xenix or Unix based multi-user systems.

AT4 – 4 users $795
AT8 – 8 users $995
AT16 – 16 users $1295

Phar Lap™ created the first tools that make it possible to develop 80386 applications which run under MS-DOS yet take advantage of the full power of the 80386. These include an 80386 monitor/loader that runs the 80386 in protected linear address mode, an assembler, linker and debugger. These tools are required for the MS-DOS version of the MicroWay NDP Compilers.

Phar Lap Tools $495

PC/AT ACCELERATORS

287Turbo-10 10 MHz $450
287Turbo-12 12 MHz $550
287TurboPlus-12 12 MHz $629
FASTCACHE-286 9 MHz $299
FASTCACHE-286 12 MHz $399
SUPERCACHE-286 .... $499

MATH COPROCESSORS

80387-20 20 MHz $795
80387-16 16 MHz $495
80287-10 10 MHz $349
80287-8 8 MHz $259
80287-6 6 MHz $179
8087-2 8 MHz $154
8087 5 MHz $99

Micro Way

The World Leader in PC Numerics

P.O. Box 79, Kingston, Mass. 02364 USA (617) 746-7341
32 High St., Kingston-Upon-Thames, U.K., 01-541-5466
St. Leonards, NSW, Australia 02-439-8400

AIS INC

421 E. Palatine Rd.
Palatine, IL 60067

312·359·2626

Software Link Inc:

Multilink Advanced®	$399
Lanlink ™ 5.0 starter kit	$349
Lanlink ™ 5.0 satellite	$79
Lanlink™ 5X Server	195
Lanlink™ 5X satellite	$85
Four port par board	CALL
AT Gizmo™	$259
PC-Emulink™	$135
PC-MOS™ single user	$135
PC-MOS™ five user	$425
PC-MOS™ 25 user	$650

ITI LinkUp products:

3270 SNA	$369
3270 BSC	$369
3770 RJE	$518

Digiboard serial boards:

4S/8S 16 bit I/O	$335/475
4I/8I Intell I/O	$678/836

Everex products:

1200b internal modem	$99
2400b internal modem	$175
2400b extrenal modem	$199
60meg tape b/u cass int	$595
60meg tape b/u cass ext	$655
40meg floppy tape b/u	$399
Magic I/O XT/AT	$65/60

TeleVideo®
Settle for more.

Terminals (new):

905 green or amber	$295
955 green or amber	$345
9550 green or amber	$339
965 green or white	$385
PC Station green	$349

Systems:
TelOAS open architecture systems
CPU upgradable — CALL

Tele-386 2m ram, 1.2fl, ser/par ports, 16mhz — CALL

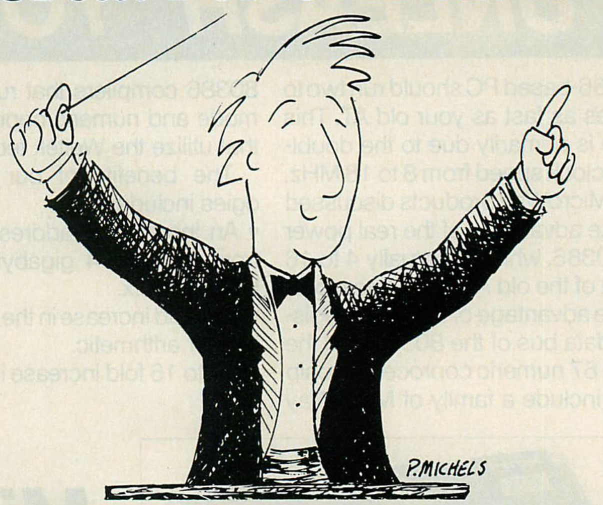
Telecat-286 model I 80286 6&8 mghz 512k ram, AT kb, mono monitor — $1695
(systems have on site warranty)

The Data Processing Professionals Dealers
(dealer inquires welcomed)

VISA MasterCard

CIRCLE NO. 131 ON READER SERVICE CARD

Great Performances



Demand an Encore

Let us take your article to its highest level of marketing potential with a reprint!

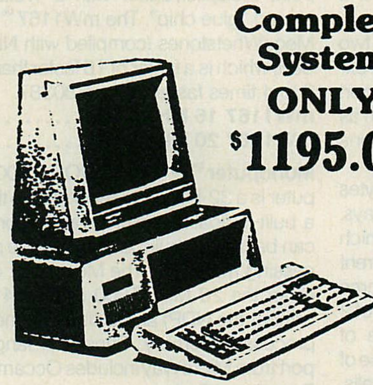
To find out how you can have your article or review elegantly reprinted** on 80 lb. paper stock, in 4-color, 2-color or 1-color, call or write today:

Jennifer Locke—Reprints Manager;
Ziff-Davis Publishing Company,
One Park Avenue, New York, NY 10016, 212-503-5447.

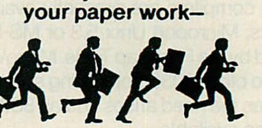
**Minimum quantity 500 reprints.

CAN YOU REALLY AFFORD TO WAIT!

Complete System ONLY $1195.00



NO! You cannot afford to wait— Don't just run from your paper work—



FACE it with confidence and take control NOW!!!

With these fabulous features and affordable prices— ask yourself — WHY WAIT? Time is money! ACT NOW!

STANDARD FEATURES

360K FLOPPY DISK	20 MB HARD DRIVE
640K RAM	1 YEAR WARRANTY-SERVICE
HGA MONOGRAPHICS VIDEO BOARD	TTL MONOCHROME MONITOR
MULTI I/O CONTROLLER CARD	XT STYLE 8 SLOT MOTHER BOARD
SERIAL PORT	PARALLEL PORT
84 KEY KEYBOARD	150 WATT POWER SUPPLY
8088-1 CPU	4/10 MHZ CLOCK SPEED
AT Jr. STYLE CASE	

CALL TODAY DON'T WAIT!

COMPUTER SOUTH, INC. 1-800-445-5930

FCC Approval CLASSB.
FEE 10 FUC4TKEDP-081

* IBM IS A TRADEMARK OF INTERNATIONAL BUSINESS MACHINES CORP.

VISA MasterCard

2614 Arlington Road
Akron, OH 44319

CIRCLE NO. 138 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PRODUCT CATEGORIES

Advertising Rates and Information: PC Tech Journal Marketplace

PC Tech Journal Marketplace is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches. Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $220.00 per issue ($660 total).

For additional information call 212-503-5115.

PC Tech Journal Classified Advertising Staff
One Park Avenue, New York, NY 10016
(212) 503-5115

Advertising Director
Kathryn J. Cumberlander

Sales Manager
Kathleen F. Lyman

Sr. Advertising Coordinator
Monica Dixon

Production Manager
Anne R. Brockinton

Stanley H. Robinson, Account Manager
(212) 503-5116
ME, CANADA – (OTHER THAN BRITISH COL.), NY, NJ, PA, MA, CT, VT, NH, RI, WV, MD, VA, NC, SC, GA, FL, DC, DE

Ariane R. Casey, Account Representative
(212) 503-5172
WA, HI, AZ, MT, NV, ID, ALL CALIF, NM, UT, CO, OR, WY

Lee J. Uniacke, Account Representative
(212) 503-5141
ND, KS, NB, OH, OK, IA, SD, KY, MI, AL, IL, LA, MN, WI, TN, MS, TX, AR, BRITISH COL., MO, ALL OVERSEAS CALLS.

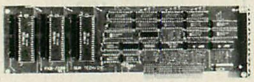
TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

HARDWARE

ACCESSORY CARDS

72 DIGITAL I/O FOR PS/2



MODELS 50, 60, 80
- Parallel Expansion
- 72 I/O Lines
- Address Selectable

1-800-553-1170

QUA TECH INCORPORATED
478 E. Exchange St., Akron, OH 44304
TEL: (216) 434-3154 FAX: (216) 434-1409
TLX: 5101012726

CIRCLE 265 ON READER SERVICE CARD

TMS 320 C25 PC COPROCESSOR BOARD NEW

- 40 MHz, 10 MIPS, Addressing 256K
- 1 Clock 16 Bit Multiply
- High Speed DSP, Graphics and Numerical Analysis
- External User Bus for A/D & D/A
- Monitor Debugger & C Utilities
- Fractals, Rotations, & FFT Demos
- Source code included

$900 board & 32K
$1100 board & 256K
$150 TMS320C25 assembler

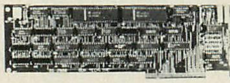
SYMMETRIC RESEARCH
15 Central Way, Suite 9, Kirkland, WA 98033
206-828-6560

CIRCLE 266 ON READER SERVICE CARD

DIGITAL SIGNAL PROCESSOR
The Model 10 coprocessor board is based on the 16/32 bit TI TMS 32010 and is designed for applications in communications, speech, instrumentation, and numeric processing. A 1K complex FFT takes 90ms. Offered with onboard 12 bit 80 Khz A/D and D/A. Continuous data acquisition & playback option. Includes all utility and applications software. $650 & up.
DALANCO SPRY
Suite 241 2900 Connecticut Ave. NW
Washington, DC 20008
(202) 232-7999

COMMUNICATIONS

RS-422 FOR PS/2



MODELS 50, 60, 80
- Two Channel
- Transfers to 256 K baud
- Address Selectable
- Interrupt Selectable

1-800-553-1170

QUA TECH INCORPORATED
478 E. Exchange St., Akron, OH 44304
TEL: (216) 434-3154 FAX: (216) 434-1409
TLX: 5101012726

CIRCLE 267 ON READER SERVICE CARD

COMPATIBLES

1988 BEST BUY

AT Bare Bones
Case, Motherboard, 200 Watt Power Supply, Phoenix Bios

10 MHZ	$465
12 MHZ	$485

XT Bare Bones
Case, Motherboard, 150 Watt Power Supply, Phoenix Bios

4.77 & 8 MHZ	$180
4.77 & 12 MHZ	$230

Min. Order - 3 ea.

80386 Motherboard ØK $1295

512-467-2330
ADVANCED MICRO TECHNOLOGY
CIRCLE 271 ON READER SERVICE CARD

OLYMPIAN DISCOUNTS!

TURBO XT BASIC SYSTEM $324
- 4.77/8 MHZ W/PHOENIX BIOS
- 256K RAM EXPANDABLE TO 640K
- AT STYLE KEYBOARD
- XT CASE w/150 WATT PS
- 360K FLOPPY W/CONTROLLER
- 10 MHZ OPTIONAL........ADD $15.00

AS ABOVE W/MONO. MONITOR & MONOGRAPHICS CARD.....$439.00
- 1 YEAR LIMITED WARRANTY

TURBO AT BASIC SYSTEM...$824.00
1200B INTERNAL MODEM........$69.00
VORTEX EGA CARD (640*350)..$99.00
ATI EGA CARD (800*600).......$189.00
SEAGATE 20MB HD W/CONTROLLER...................$265.00

CALL FOR FULL PRODUCT LINE
Quantity Discounts Available
OLYMPIA MANUFACTURING, INC.
800-527-0806
CIRCLE 270 ON READER SERVICE CARD

COMPATIBLES

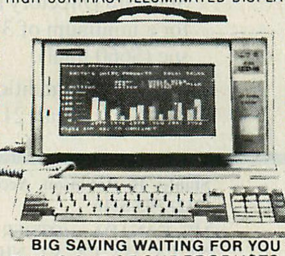
AdvanTech Offers 286-16MHz Power — ABSOLUTELY!!!

Announcing our VHP/AT–16 w/TRUE ZERO WAIT STATE
- ❒ Very Latest THREE (3) AT Chipset Technology
- ❒ 1Mb of superspeed DRAM on board
- ❒ ABSOLUTE–16/0 AT motherboard now available in qty.
- ❒ Proprietary design for fastest 80286 to date
- ❒ Out performs many 80386–based systems

System boards from *$895* Tower systems from *$1795*
1-800-338-3130

AdvanTech Corporation 261 Cedar Hill Street Marlborough, MA 01752 (617) 481-6009
CIRCLE 268 ON READER SERVICE CARD

Portable AT/386
HIGH CONTRAST ILLUMINATED DISPLAY
IBM COMPATIBLE
QUALITY, SERVICE



BIG SAVING WAITING FOR YOU
THIS IS ONE OF OUR PRODUCTS
ASK FOR LATEST CATALOGUE
MAXTRON (818) 350-5707
1825A Durfee Ave., S. El Monte, CA 91733
CIRCLE 269 ON READER SERVICE CARD

To Place Your Ad
Call (201) 503-5115

PERIPHERALS

New! SPEECH THING™
For all PCs, compatibles, laptops

$69.95 CLEAR, CLEAN, DIGITIZED SPEECH & MUSIC, UNLIMITED TEXT-TO-SPEECH
Attaches outside the computer.

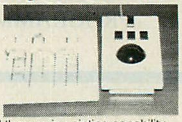
Now get popular speech technologies in ONE product! Speech Thing is a full-featured 8 bit D/A sound converter. Easily attaches outside the computer–no slots required. Software includes prerecorded speech vocabularies, synthetic text to speech (speaks any ASCII text), demo programs, and complete editing features. Price only $69.95. Also available: Voice Master PC plug in board for digital recording, editing, and VOICE RECOGNITION. (Requires Speech Thing for sound output.) Only $79.95. Patented price/performance breakthroughs!

TO ORDER BY MAIL include $4 shipping & handling ($6 Canada, $12 overseas) per order. Visa, MasterCard phone orders accepted. 30 day money back guarantee, one year warranty. Other voice I/O systems available for Apple, Commodore, and Atari computers.

Call or write today for FREE Product Catalog
COVOX INC. (503) 342-1271
675 Conger St., Eugene, Oregon 97402

CIRCLE 272 ON READER SERVICE CARD

FastTRAP™
The pointing device of the future is here!



- Two and three axis pointing capability
- High resolution trackball for X and Y axis input
- High resolution fingerwheel for Z axis input
- Use with IBM® PC's, XT's, AT's and compatibles
- Three input buttons
- Full hardware emulation of Microsoft® Mouse
- Standard RS-232 serial interface
- Includes graphics drivers and menu generator
- Easy installation
- 1 year warranty
- Made in the U.S.A.

ONLY $149.00
VISA and MasterCard accepted

LTS/C Corp.
319 South Limestone Street
Lexington, Kentucky 40508
(606) 233-4156
(800) 872-7279

3-D TRACKBALL FOR IBM
CIRCLE 273 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

Peripherals

"NOBODY"

We asked our customers to tell us who had LOWER PRICES than we do.

They told us!

We lead the industry both in price and quality in ½" 9-track tape drive subsystems to transfer information between mainframes and PC/XT/AT, PS2.

- EBCDIC-ASCII Conversion
- Tape backup/volume data stg.
- Speeds up to 7 Meg/min.
- Select & reject specific fields
- 800/1600/6250 BPI
- Drives from Anritsu, Cipher, Qualstar and M4 Data

Dealer & Volume Discounts

FLAGSTAFF ENGINEERING

1120 Kaibab Lane • Flagstaff, AZ 86001
(602) 779-3341 • Telex 705609
FAX (602) 779-5998

CIRCLE 274 ON READER SERVICE CARD

NOVELL NETWORKING PRODUCTS

Workstations 286, 86
3 station network $4500 and up
Disk subsystems:
- 109Mb–$1795; 183Mb–call; 244Mb–call

Add-on drives for NDS-2, NDS-4
- 109Mb–$1350; 183Mb–$2650; 244Mb–call

Call for other networking products.
NETWORKING PRODUCTS, INC.
114 West Stratford Avenue
Lansdowne, PA 19050
(215) 630-9746

SPEECH SYNTHESIS

SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix
Electronic Speech Articulator

Artic Technologies
55 Park St., Suite 2
Troy, MI 48083
(313) 588-7370

CREATE A DISKLESS PC!

PC-ROMDRIVE allows users to create a "Diskless PC" capable of booting a ROM-resident copy of MS-DOS and/or user application programs. PC-ROMDRIVE consists of a PC-compatible ROM/PROM expansion board and the PC-ROMDRIVE software. PC-ROMDRIVE is priced at $195 for single units. Quantity discounts and OEM arrangements available. MC/VISA
ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, Az. 85069
(602) 866-1786

TECH MARKETPLACE
HOME OF THE
POWER BUYER

Software

Business

LP88-SPREADSHEET LP

Our best-selling linear programming system solves problems w/1000 constraints and 5000 variables, reads/writes Lotus worksheets, many other advanced features. IE News says "the flexibility and features of this program are a bargain at its low price." Req. 192k. $149 w/manual and 8087 support. $29 demo.
Eastern Software Products, Inc.
P.O. Box 15328
Alexandria, VA 22309
(703) 360-7600

Communications

DOS COM PORT DEVICE DRIVER

Interrupt driven device driver used for the COM ports on PC, AT & compatibles. With source code in Assembly & test program in C. Use instructions as open, close, read, write & ioctl to access your COM port data. Load driver through config.sys at boot up. $39.95 with manual. s/h + $5.00. CA res. add $2.60 tax.
HYTEC RESEARCH, INC.
22324 Harbor Ridge Lane, #4
Torrance, CA 90502
(213) 320-4541

Tech Marketplace...
The comprehensive guide to products and services for the MS DOS market.
To place your ad
Call
(212) 503-5115

Data Base Management

RT-DBMS

Real-Time Data Base Manager for Fortran applications

EFFICIENT - Transfers 10,000 records/second (80286 & hard disk).
ADVANCED - Customize data structures. Program object-oriented software. Use table driven methods.
SIMPLISTIC - Data transfers use the Fortran "Common" Block storage.
HELPFUL - RT-DBMS controls data typing, storage and "Common" blocks.
FLEXIBLE - Break large software modules into small tasks, passing data via RT-DBMS.
SUPPORTS - Microsoft, Ryan-McFarland and Lahey Fortran. "C" version available soon!

Includes:
Royalty Free Run-Time Manager
Data Base Manager
Data Base Editor
Data Archiving Manager

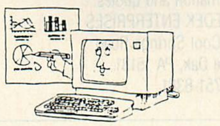
$145.00 includes shipping
use VISA & MASTERCARD

Ashley Software Engineering
18219 228th Ave N.E.
Woodinville, WA 98072
(206) 788-3768

CIRCLE 275 ON READER SERVICE CARD

Graphics

PlotIT®



Interactive Graphics & Statistics

Create graphs the way you want them. 35 graph types can be used to create numerous renditions. Make posters and flow charts—with full screen editing of text.

- Vary character font, size, position, color
- Stack and overlay graphs
- 3-Dimensional graphs: fishnet & contour
- Linear & Non-Linear Regression
- Stochastic Distribution
- Independence Testing
- Descriptive Statistics

Contact: Scientific Programming Enterprises, P.O. Box 669, Haslett, MI 48840 (517) 339-9859

CIRCLE 276 ON READER SERVICE CARD

Graphics

DO YOU NEED SPEED?

W. D. Bowles

It took nearly 5 minutes to print 10 of these signatures, using a well-known word processor and its internal graphics merge.
But with the same word processor and **SLEd,** it took less than 3!

GRAPHICS AS FAST AS A FONT!

SLEd offers a remarkably simple method of incorporating scanned images (photos, logos, signatures, etc.) into your documents using only your word processor and laser printer **$149.95**

VS SOFTWARE
P.O. Box 6158
Little Rock, AR 72216
501-376-2083

CIRCLE 277 ON READER SERVICE CARD

FORTRAN Graphics Libraries

Now! 3 scientific graphics packages to support MS FORTRAN/Pascal, R-M or Lahey FORTRAN. GRAFMATIC (screen), PLOTMATIC (HPGL, H-I plotter) & PRINTMATIC (Laser & dot-matrix printer). All fully documented. Complete graphics primitives, 2-D plots, 3-D plots and solid models. $135 each, 2 for $240, all 3 only $340. Call or write for info.
Microcompatibles, Inc.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Use high resolution color slides up to 4000 line. COMPUTER SLIDE EXPRESS $9/slide.
VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

GRAPHICS
Category
continued on next page

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

GRAPHICS

PLOTTER EMULATION (w/VGA)
FORTRAN callable, Versatec/Calcomp compatible plot subroutines for VGA, EGA, CGA, Herc. video and Epson and compatible printers at optimum resolution. Contour and 3d plots. 13 character fonts. Libraries for MS 3.3/4.0 and Lahey F77L, 150 page manual with examples. Only $200, HP plotter driver $50. Educational discounts.
F and S Software
7604 Peacock Drive
Huntsville, Alabama 35802
(205) 881-6268

GIPS Image Processing
Comes complete with manual and tutorials. Runs on PCVision Plus, Number Nine and AT&T VISTA boards. Features include basic operations, geometry, edge and line detection, texture analysis, etc. Prices: GIPS system $550, GIPS feature Analysis (microscope) $550 GIPS Sourcecode $5000.
PC-Expand, Inc.
St. Kannikestraede 7
DK 1169 Copenhagen K, Denmark
Tel. +45 1 156800, FAX +45 1 939901

********ADULT SOFTWARE********
Watch these girls! Let your PC/XT/AT, Compatibles & you have some real entertainment. Games, graphics and more for adults only. Volumes 1 and 2 are $11.95 each ppd. Volume 3 $16.95 ppd. Confidentiality assured. Color or graphics card needed. Must state and sign age as over 21.
A.O.S.
P.O. Box 106
Marion, CT 06444

TECH MARKETPLACE...
Standard directory listings available.
To place your ad
Call
(212) 503-5115

NETWORKING

PRINT BUFFER/SPOOLER utility. 4 serial & 3 parallel ports. Local/remote control. To 38400 BAUD. Forms merge. $59.95 (Foreign +$5)

TRUE MULTI-TASKING toolbox for Turbo Pascal. Fast & Easy. Demand & Preempt. $89.95.‡

MULTI-PROCESSING toolbox for Turbo. (w/ NET BIOS supt) Modem/RS232 network. $149.95

‡ +$5 shipping (+$10 foreign) In Tx add Tax. VISA, M/C, COD. For more info call or write:

Conversational Computer Systems
5371 Verbena Rd
San Antonio, Tx. 78240
Ph: (512) 692-0353
CIRCLE 279 ON READER SERVICE CARD

KEYCARD ELIMINATOR

NOVELL ADVANCED NETWARE®
V2.0a
86, 286, 286 nonded, SFT-1, SFT-2

SLOTBOUND?
FREE A SLOT IN YOUR FILESERVER!
M/C, Visa or C.O.D.
$99.00 (includes S & H)
Tx Residents add 7.25% Tax
CALL FOR DEALER PRICING
NBS
NETWORK BUSINESS SYSTEMS
1300 Woodhollow Drive, Suite 5601
Houston, Texas 77057
713-781-9268 (Sales, Tech Support)
713-783-4457 (Administrative, Sales)
* NetWare is a Trademark of Novell, Inc.
CIRCLE 281 ON READER SERVICE CARD

Run Novell Without Keycard
Protect Novell Network From Keycard Failure
Netsafe For 2.x $99 U.S.
Other products for P-Cad, Cadkey, Cadvance
VISA AMEX MASTERCARD
Nasdec
NASDEC (MB) INC.
72 - 125 Garry Street
Winnipeg, Manitoba,
Canada R3C 3P2
(204) 956-2798
CIRCLE 280 ON READER SERVICE CARD

NETWORK CONTROL LIBRARIES
NETWORK INTERFACE allows file sharing and redirection through DOS functions. $99. NETBIOS ROUTINES allows access to low-level network functions. Name, session & datagram routines. Wait and no-wait options. $199. NETWORK MASTER provides access to Netware internal functions. Complete control of your network from your compiled programs.
Starlight Software
2821 Central Street
Evanston, IL 60201
(312) 864-9370

NOVELL NETWORKING PRODUCTS
When reliability and performance really count you can't afford anything less than Novell. We offer quality Novell products and accessories, fast courteous service and competitive pricing for all your networking needs. Our experienced staff is waiting to serve you. Call today for information and quotes.
BENEDEK ENTERPRISES
104 Cool Springs Road
White Oak, PA 15131
412-751-8381

PROGRAMMERS TOOLS

MS-DOS for VAX
- DEVELOP PC applications
- DEBUG PC code
- RUN as if on a PC
- All as a VMS process

$795
M software inc
708 W. Huron
Ann Arbor, MI 48103
313/761-5800
CIRCLE 282 ON READER SERVICE CARD

'C' DOCUMENTATION TOOLS
SAVE hours of debugging and documentation! Automatically Document the internal structure of 'C' programs with easy-to-use **"C-DOC"** tools:
- **C-CALL** ($39) produces system/module level XREFs and graphic trees of all caller/called (flow) structures within a group of 'C' programs.
- **C-REF** ($29) produces system/module XREFs of variables/constants (incl. global vs local).
- **C-HDR** ($19) automatically uses the above 2 tools to generate/modify module headers.
- **C-LIST** ($29) lists and diagrams 'C' programs or reformats them (incl. comment positions).
- 30-day money-back guarantee of satisfaction.

SPECIAL!! All 4 programs ($126) for only $89
SOFTWARE BLACKSMITHS INC.
6064 St Ives Way,
Mississauga, ONT, Canada. L5N-4M1
(416) 858-4466
CIRCLE 283 ON READER SERVICE CARD

PROGRAMMERS TOOLS

ROM DEVELOPMENT TOOLS
Link MS-DOS and Intel object files to ROMable code. GeneLink permits independent placement of each program segment for complete flexibility in memory layout. GeneLink links directly from object files to HEX or other formats, and is five to ten times faster than other locating tools.
Genesis has been a major supplier of ROM development tools to major industrial customers like IBM and Boeing for over five years. You can count on Genesis to provide reliable, fast tools and excellent support.
Genesis Microsystems
Genesis Microsystems Corporation
196 Castro St., Mountain View, CA 94041
Call: (415) 964-9001; TX: 4998093 GENMS UI
CIRCLE 284 ON READER SERVICE CARD

NETWORKING

* MULTI USER / MULTI TASKING *

PC-MOS/386		WYSE 60 W/KEYBOARD	500.00
SINGLE USER	160.00	WYSE 99GT W/KEYBOARD	600.00
5 USER	510.00	AT GIZMO	300.00
25 USER	910.00	+4 SERIAL BOARD	470.00
PC EMULINK	160.00	+8 SERIAL BOARD	570.00
		+4 INTELLIGENT SER/BOARD	640.00
LANLINK (COMPLETE)		+8 INTELLIGENT SER/BOARD	840.00
5.01	410.00	4 USER 286 NETWORK (COMPLETE)	6595.00
5X	525.00	4 USER 386 NETWORK (COMPLETE)	8795.00

FLA RESIDENTS ADD 6% SALES TAX — CALL FOR DETAILS
BUSINESS DEVELOPMENT SYSTEMS (800)922-8147
SHIPMENTS - 2ND DAY AIR / CASH|CASHIER'S CHECK
DEALER INQUIRIES WELCOME
CIRCLE 278 ON READER SERVICE CARD

Put Workstation-Quality Graphics in Your PC User Interfaces
SKYLIGHTS™ lets you have windowing user interfaces without buying graphics hardware. If you have graphics hardware, it lets you have high-resolution, workstation-quality graphics --- far beyond WINDOWS.

SKYLIGHTS
- Unlimited look and feels.
- Easy prototyping of screen designs.
- Supports mice, lightpens, trackballs, digitizers, tablets, Hercules, CGA, EGA, VGA, Tecmar, TARGA-16, major C compilers, all IBM and compatible PCs and PS/2 models.
- Reduces system development time by 50% or more.
- Interactive screen building --- no coding necessary.
- Import screens from CAD applications, paint programs and prototypers.

Character-mode only, $295. With graphics, $750. For more information or a demo disc contact:
"**SKYLIGHTS** paid for itself in the first week. All the other interface development tools out there are just toys compared to **SKYLIGHTS**." Jerald Feinstein, Vice-Pres, PLS/ICF, Wash. DC.
Ergosyst Associates, Inc.
910 Massachusetts St., Suite 602PCT
Lawrence, KS 66044 (913) 842-7334
CIRCLE 285 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

C & MASM TOOLS

VersiMAKE™
A full-featured MAKE utility that derives your system's dependencies through analysis of your C & MASM source files. No more MAKE dependency files to maintain!

VersiCREF™
A unique utility that creates a sorted Master Cross-Reference of your entire system. Handles 100+ C and MASM source files. Full X-Ref or just PUBLIC symbols.

VersiMAKE™	$125
VersiCREF™	$75
Both	$150

800-334-4096
(In NJ, 609-871-0202)
MC/VISA/AMEX

SUMMIT INFORMATION SYSTEMS, INC.
73 East Lane, Willingboro, NJ 08046
CIRCLE 293 ON READER SERVICE CARD

MULTI-TASKING TOOLKIT

Write your "C" programs with cooperating, concurrent tasks using INTERWORK™. Very useful for parallel programming, simulation, and real-time applications. Features:

- more than 100 tasks
- inter-task communication facilities
- DOS interrupt handling

Documentation includes user's guide and reference manual. PC-DOS version $129, Xenix $159, Unix $249; source code $995. Shipping included; UPS 2 day air add $5. For more information, contact:

Block Island Technologies
Innovative Computer Software

13563 NW Cornell Road, Suite 230
Portland, Oregon 97229 (503)241-8971
CIRCLE 295 ON READER SERVICE CARD

SOURCER™

Create detailed commented source code and listings from memory, .COM files or .EXE files directly suitable for assembly. Built in data analyzer and simulator resolves multiple data segments and provides detailed comments on interrupts and subfunctions, I/O ports and much more.

BIOS SOURCE

PS/2 ▪ AT ▪ XT ▪ PC ▪ Clones

The bios pre-processor to SOURCER™ provides the first means to obtain accurate legal source listings for any bios! Identifies entry points with full explanations. Provides highly descriptive data labels such as "video_mode" and much more.

SOURCER	$ 99.95
SOURCER w/BIOS Pre-Processor	$139.95

To order or receive information just call!
800-538-8157 x 811 (outside Calif.)
800-672-3470 x 811 (inside Calif.)

V COMMUNICATIONS
3031 Tisch Way, Suite 200, Dept. TJ
San Jose, CA 95128 • (408) 296-4224

PS/2, AT, XT, PC are trademarks of IBM Corp.
CIRCLE 296 ON READER SERVICE CARD

SCREEN MANAGER

MENU, WINDOW, and DATA ENTRY Support for the Professional Programmer! Interfaces to most languages. BASIC, C, FORTRAN, COBOL, PASCAL, ASSEMBLER. 100 Page Manual. Thirty day money back guarantee. No Royalties.

from **$79**
VISA/MC

The West Chester Group
P.O. Box 1304
West Chester, Pa 19380
(215) 644-4206

CALL FOR FREE DEMO
CIRCLE 294 ON READER SERVICE CARD

Intelligent Front End™

DataPlex 1.0® $149

New BREAKTHROUGH data entry and file conversion S/W – PC/XT/AT & full compatibles.

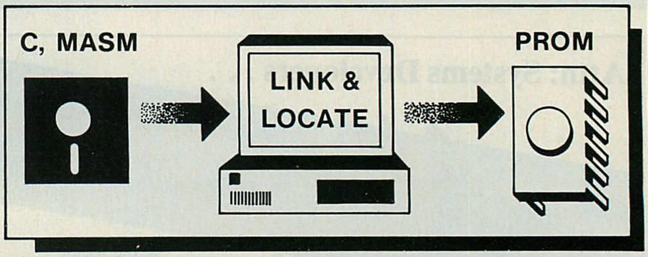
- **Super-fast** keyboard data entry to any destination
- **Automatic** field formatting and extensive validation
- **No programming** involved. Eliminates screen input forms!
- Consistent access = **savings**
- Converts with intelligent filtering **dBASE, 123, ASCII, PARADOX, EBCDIC, DIF, SC4, FW II, ENABLE,** & more!

Tools & Techniques Inc. 512-482-0824
1620 W 12th St., Austin, TX 78703
800 - 444 - 1945
MC/VISA/COD/PO
CIRCLE 297 ON READER SERVICE CARD

Easy Assembly Maintenance

Debug, develop and maintain assembly language faster and easier with PAL structured concepts, IBM/MASM compatible. Use more powerful commands and new verbs, including: If, Select, Do, Leave and Include. $50.
LANEY SYSTEMS, INC.
3 Office Park Drive, Suite 100
Little Rock, AR 72211
(501) 225-7755

FIRMWARE DEVELOPMENT



LINK & LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by popular C compilers, such as from Microsoft, Lattice and Borland's Turbo C, and MASM from Microsoft, Provides full control of segment placement anywhere in memory. Supports output of Intel HEX file for PROM programmers, Intel OMF absolute object file for symbolic debuggers and in-circuit emulators. Includes Intel compatible linker, locator, librarian, hex formatter and cross reference generator. $350.

NEW! Includes utility to support PC based PROM programmers.

SYSTEMS & SOFTWARE INC.
3303 Harbor Blvd., C11, Costa Mesa, CA 92626
Phone (714) 241-8650 FAX (714) 241-0377 TWX 910-695-0125
CIRCLE 298 ON READER SERVICE CARD

COMPLEX FILE I/O PROBLEMS?

Heap I/O: A C library that can allocate, free (in random order), read and write variable length blocks of storage in a file. Use it to implement VM swappers, editors, databases, and other systems that require complex, varying length disk I/O. Microsoft C object $59.95. For source add $79.95.
ROLAND ALDEN SYSTEM SOFTWARE
One Pine Street, Suite 2509
San Francisco, CA 94111
(415) 397-9316

GRAPHIC CHARACTERS I/O

Screen messages for your software in any graphics mode. Characters that can be varied in size and color on the fly. A called subroutine for Quick Basic and the IBM Basic Compiler. $29.95 + s/h for object module or $59.95 + s/h for source module.
Com Tech Systems, Inc.
P. O. Box 968
Blue Springs, MO 64015
816-228-5239

Affordable CASE

A new concept in Computer Aided Software Engineering for developing PC/DOS applications! C Dispatcher generates fast, efficient C code for command and menu driven app's. Develop, document, and change easily. Many features. For many compilers.
Amaryllis, Inc.
563 Wattaquadoc Road
Bolton, MA 01740
(617) 365-5456

"Better Than Bricklin"

Use the best prototype/demo system available today. Proteus makes it easy to create modular, fully-interactive presentations. Also draw program screens, forms and charts. Menu-driven, context-sensitive help, complete documentation. 30-day guarantee. $99. VISA/MC/COD/PO.
Helios Software
P.O. Box 22869, Seattle, WA 98122
(800) 634-9986 (206) 324-7208

TURBO WINDOW TOOLBOX

A complete windows programming environment for Turbo Pascal. Popup windows, pulldown menus & messages, virtual screens and a multi-window WordStar-like editor. Animated resizing, moving, horizontal & vertical scrolling with blinding speed. Optional mouse. All source code and a 90+ pp. manual provided.
Turbosoft
369 6 Avenue
Brooklyn, N.Y. 11215
(718) 965-9729/No royalties $50 checks only

PASCAL-TO-C TRANSLATOR

Convert to UNIX and OS/2 using our 98% automatic translation tools: units, strings, denesting, all types, modules, I/O. The most complete translators for Turbo, MT+, Microsoft, UCSD, Apollo, Macintosh, others. Industrial strength licensing from $7,500 includes technical support, source code libraries, and IBM PC/AT executables.
(503) 745-7476 or 5880
TGL, INC.

TECH MARKETPLACE...

Second Color Option Available
Call: (212) 503-5115

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

Programmers Tools

TURBO PASCAL 4.0 Units
Control MS mouse from Pascal. On screen mouse cursor builder included. $39.95
Interface to dBase II/III. DBF files. Query, append & update; multiple tables & virtual fields. $74.95 Both for $99. $2.50 s+h per order. Money order, cashiers or personal check.
Publishing Support Services
9222 NE 139St.
Kirkland, WA 98034
206-998-0035

Public Domain

PROGRAMMER'S CHOICE!
Don't look any further for quality public domain & shareware programming utilities. The Complete Basic, or The Complete Pascal set, each contain 5 disks packed with the most popular and useful utilities to date. Each set, Pascal or Basic just $12.95 plus $2 shipping. N.J. residents add $.79 tax. Make Check or Money Order payable to:
Brimstone
P.O. Box 1653
South Hackensack, N.J. 07606.

Public Domain Software in C
Over 150 volumes of public domain software for MSDOS, UNIX and CP/M.
- small expert systems and graphics
- editors, compilers, text formatters
- many UNIX-like tools & misc. utilities

Write or call for more details. Send $10 for comprehensive directory.

C Users' Group

The C Users' Group
PO Box 97
McPherson, KS 67460
(316) 241-1065

Scientific

SCI/ENG GRAPHICS
OMNIPLOT [S] (screen graphics) & OMNIPLOT [P] (plotter driver) provide integrated engineering/scientific 2-D & 3-D graphics with NO PROGRAMMING! Menu-driven, flexible, professional. Choice of formats: tabular/line, contour, bar, pie, 3-D wire frame & much more! OMNIPLOT [S] $195. Add OMNIPLOT [P], both $295.
MICROCOMPATIBLES, INC.
301 Prelude Dr. Dept. J
Silver Spring, MD 20901
(301) 593-0683

Tech Marketplace
Standard Directory
Listings Available
Call (212) 503-5115

Security Devices

EVERLOCK COPY PROTECTION

Designed for user-transparency, clone compatibility & strength. It features:
- no need for damaged media or I/O plugs
- supports all Hard & Floppy disk formats
- file-server networks supported
- variable number of installs (0-99)
- create demos with remote unlock option
- allows protected upgrades by modem/BBS

$195 Starter Kit or $495 with NO meter counts. Free info & demo disk available.

Az-Tech Software, Inc.
305 East Franklin
Suite A4A
Richmond, MO 64085
(800) 227-0644 . . . (816) 776-2700
CIRCLE 300 ON READER SERVICE CARD

BIT-LOCK® SECURITY
Piracy SURVIVAL 5 YEARS proves effectiveness of powerful multilayered security. Rapid decryption algorithms. Reliable/small port transparent security device. PARALLEL or SERIAL port. Countdown and timeout options also available. KEY-LOCK™ security at about ½ BIT-LOCK cost.
MICROCOMPUTER APPLICATIONS
7805 S. Windermere Circle
Littleton, CO 80120
(303) 922-6410 or 798-7683

Terminal Emulation

BARR High-Performance RJE Workstations

BARR/SNA RJE and BARR/HASP are communications hardware/software packages for the IBM PC and PS/2 series of computers.

Full featured:
- multiple printers,
- unattended operation,
- print speed beyond 6,000 lpm,
- special forms,
- line speed to 56,000 bps,
- dial-up or dedicated lines, and
- serve as a LAN gateway.

BARR/SNA RJE emulates IBM 3777-3 in an SNA environment. BARR/HASP emulates IBM 3777-2 and HASP on the 360/20. BARR communications software is also available with PC-SYNC internal modems: 208AB, 201C, 9600.

BARR

Barr Systems, Inc.
2830 NW 41 St. Bldg M
Gainesville, FL 32606
800-BARRSYS or 904-371-3050
CIRCLE 299 ON READER SERVICE CARD

Utilities

How to get through traps with

Quaid Analyzer

Quaid Analyzer is a powerful diagnostic tool that shows what is going on inside your computer. Since we developed **Quaid Analyzer** to unravel copy-protection, we had to include ways of tracing in spite of the countermeasures used to block us. Some common ones are:

Overwrite the vector for interrupt 3, so you can't use breakpoints. **Quaid Analyzer** can use any of the other 255 vectors for breakpoints instead.

Overwrite the vector for interrupt 1, so you can't single step. **Quaid Analyzer** scrupulously saves and restores interrupt cells when changing them. You won't notice changes to vector 1 except at the one instruction doing the change. Step over it with another command.

Checksum the program to detect breakpoints. **Quaid Analyzer** can report every interrupt call. Since DOS programs have to call interrupts for all system services, you get control frequently, and at revealing places. Single stepping is unaffected by checksums.

Manipulate the t flag to stop single stepping. Just avoid the single step key for one instruction. **Quaid Analyzer** has two ways to step over a few instructions.

Point the stack over the interrupt vectors. Just ask **Quaid Analyzer** to catch the next system service call interrupt. The program will have to fix everything first.

Of course, your own program probably doesn't have any tricks to block tracing. With a tool as powerful as **Quaid Analyzer**, it should be a simple matter to diagnose it.

Quaid Analyzer comes with a manual, and software on a 3 inch and a 5 inch diskette. If you are not satisfied with **Quaid Analyzer**, you can return it within 30 days for a refund. **Quaid Analyzer** is not sold by dealers in the United States or Canada. It is not copy-protected.

To order **Quaid Analyzer**, call us with your credit card, or send us a check for $200 US funds. We ship within a day at our expense.

Quaid Software Limited
Third Floor Dept T630
45 Charles Street East
Toronto Ontario Canada M4Y 1S2

(416) 961-8243

Warning! For advanced programmers only.

CIRCLE 375 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

UTILITIES

Compress Your Data 10X FASTER!!

- PKARC & PKXARC can significantly **INCREASE DISK STORAGE CAPACITY** and reduce file transfer times!
- PKARC & PKXARC can compress your files even smaller and up to TEN TIMES FASTER than the other ARChive program.
- Data encryption capability too! Not copy protected.
- **"PKARC/PKXARC is the system to use."** -Dr. Dobbs Journal of Software Tools
- **"Lightning-fast"**, INFOWORLD

Only $45 + 3.50 s/h. WI res. add 5% sales tax.
7032 N. Ardara Ave., Glendale, WI 53209 (414) 352-3670

PKware Inc.

CIRCLE 376 ON READER SERVICE CARD

AT's DON'T NEED 360KB DRIVES

The 1.2MB drive has long been known to **READ** but **NOT** reliably **WRITE** on 360KB floppies. With **"CPYAT2PC"** 1.2MB drives **CAN** reliably **WRITE** 360KB floppies saving a slot for a second hard disk or backup tape. **"CPYAT2PC"** (Not Copy Protected) offers the preferable **SOFTWARE SOLUTION.**

- **NO** software or hardware modification
- A 360K drive is **NOT** required
- **"CPYAT2PC"** program **MAY** reside on hard disk
- Runs on IBM PC/AT and COMPATIBLES i.e. Compaq Deskpro 286/386, AT&T 6300+, HP Vectra, Sperry PC/IT, Tandy 3000

Only $79.00 + $4.00 S/H VISA, MC, COD, UPS-B/R
ORDER TOLL FREE 1-800-523-8777
TELEX EZLINK 62873089
Dealer Inquiries Invited
MICROBRIDGE COMPUTERS
655 Skyway #125
San Carlos, CA 94070
CA 415-593-8777
NY 212-334-1858

CIRCLE 377 ON READER SERVICE CARD

MICROCACHE™ and SPEEDPRINT™

See our product review in this issue!

MICROCACHE™ monitors disk use and "learns" to anticipate your data accesses. Then, by satisfying these requests directly from RAM instead of disk, your programs run as if they are SUPERCHARGED.

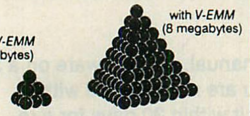
- 100% Transparent Operation
- Uses any spare RAM available
- EMS/EEMS memory support
- SPEEDPRINT™ buffer option
- Requires NO program changes
- For IBM and compatibles

$89.00 Reduced to $69.00

Az-Tech Software, Inc.
305 East Franklin
Richmond, MO 64085
(800) 227-0644 . . . (816) 776-2700

CIRCLE 380 ON READER SERVICE CARD

Ammunition to conquer your biggest problems!



- The **Virtual Expanded Memory Manager** provides your EMS applications with up to eight megabytes of expanded memory.
- It turns PC/AT extended memory into expanded memory.
- It turns hard disk storage into expanded memory.
- It's compatible with most EMS applications, including 1-2-3, Symphony, and Reflex.
- *Only $89.95. 30-day money-back guarantee.*

V-EMM™

Fort's Software
P. O. Box 396
Manhattan, KS 66502
(913) 537-2897

CIRCLE 378 ON READER SERVICE CARD

COMPRESS YOUR DATA!

ARC is considered the industry standard archive utility. It's the file storage method used by PCTECHline. And PC WEEK called it "A sophisticated and eminently useful product."

ARC automatically compresses stored data so it takes up less space. And less modem transfer time. From 20% to 90% less, depending on the kind of data.

All program sources are included and it's not copy protected .... **$50 ARC**

System Enhancement Associates
21 New Street, Wayne, New Jersey 07470•(201) 473-5153

CIRCLE 379 ON READER SERVICE CARD

TECH MARKETPLACE . . .

The Home Of The Power Buyer

HARD DISK EXPANSION!

Disk Manager allows the installation of any ST506 hard disk on PC,XT,AT and compatibles. Volumes up to 256mb! Menu driven/auto install, compatible w/ all vers of MS/PC DOS (does not modify DOS), up to 16 volumes, easy to use! $125 + ship. Ask about Novell product! Dealer inquiries invited.

ONTRACK COMPUTER SYSTEMS INC.

ONTRACK COMPUTER SYSTEMS, INC.
6222 Bury Drive
Eden Prairie, MN 55344
(612) 937-1107 1-800-752-1333

STILL RIVER SHELL, New Ver 2.44

*ELEGANT INTERFACE TO DOS Faster more powerful version of this popular DOS shell. Find, copy, move, delete files in a few keystrokes. *Tagging *Find on name and text *Full Tree *User defined commands and more. Unltd. site lic. $390. $39 complete with 160pp. manual. MC/VISA. 90-Day No-Risk Guarantee.
Bill White
P. O. Box 57
Still River, MA 01467
(617) 456-3699

XT/AT HARD DISK DIAGNOSTICS!

Disk Manager Diagnostics performs extensive tests on your ST412/506 hard disks. Areas tested are: Controller, data write/read, seek test, automatic error correction (ECC), random reads and media defects. Interactive help. Excellent error detection and isolation. $49.95 + ship. VISA/MC accepted.

ONTRACK COMPUTER SYSTEMS INC.

ONTRACK COMPUTER SYSTEMS, INC.
6222 Bury Drive
Eden Prairie, MN 55344
(612) 937-1107 1-800-752-1333

DISK ACCELERATOR V2.0

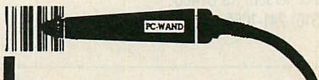
DiskCache speeds up your hard disk access. Disk caching and ram disk in one package. Ram disk shares cache space. Transparent, flexible, configurable, no h/w changes. RAM, EMS, and AT extended memory versions incl. Not copy protected. VISA, MC, volume discounts. No PO's w/o prior approval. $49.00
DATAMORPHICS LTD.
P.O. Box 820
Stittsville, Ontario, Canada K0A 3G0
Or call (613) 831-0409

An Intelligent Hexdump

VF lets you view, edit, and "debug" C, Pascal and BASIC data files. No more "HEXDUMPing" databases: VF converts int's, long's, floats, labels. You can search within fields and follow pointers between records. OS/2, MS-DOS compatible. $50.00 (TX residents $53.75).
MetaMicro Library Systems, Inc.
6011 Broadway, Suite 203
San Antonio, Texas 78209
(512) 829-7033

MISCELLANEOUS

BAR CODING



BARCODE BARGAINS

NEW - WANDERWAND™ BARCODE READER

Think of everything you would ever want in a barcode reader and you'll find it in the new ITS 5308 system! Programmable • Handheld, lightweight • Simple connection with standard RS 232 plug • Holds up to 48,127 characters • Discriminates between label codes, time stamp labels • Calculates and dumps scans • Wand included

ITS 5306 - PC Wand Barcode Reader. Emulates keyboard • Reads codes UPC, Codabar, 3 of 9, 2 of 5 interleaved • Works with IBM PC's and clones • Lowest cost - versatile, high quality!

PCDOS-MSDOS Printing Software. Generate your own labels on your existing Matrix printer • Call for details.

International Technologies & Systems Corp.
1950 White Star Dr., Diamond Bar, CA 91765
PHONE: 1-(714) 861-7977 • TELEX 6502824734 MCI

GUARANTEED LOWEST PRICES
Visa and MasterCard Accepted
ITS

CIRCLE 381 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

BAR CODING

BAR CODES MADE EASY

PERCON® E-Z-READER™

FAST • ACCURATE • RUGGED

NO SOFTWARE CHANGES with PC/XT/AT and PS/2, AT&T 6300/7300, Wyse 30/60/85/PC/AT, Kimtron KT-7/PC, Link PCTerm/MC1/MC3, TeleVideo PCS1/905/955 & DEC VT220 keyboards, multi-user RS-232 interfaces.

- Immediate shipment
- Free phone support
- 2-year warranty
- Bar code printing software available

Details or Questions? Call us.

(503) 344-1189

2190 W. 11th Ave., Eugene, OR 97402

A LEADER IN BAR CODE READER ENGINEERING

PERCON®

BAR CODE & MAGNETIC STRIPE READERS for the IBM PC & PS/2

Simple & quick installation
No additional software or port
Metal wand & case

Also available:
Bar code printing software
Magnetic stripe encoders
Units for other computers & terminals
GSA pricing available

TPS ELECTRONICS
4047 Transport Street
Palo Alto, CA 94303
Telephone: 415-856-6833
Telex: (Graphnet) 371-9097 TPS PLA
FAX: 415-856-3843

CIRCLE 383 ON READER SERVICE CARD

READ AND PRINT BAR CODES

Fast, reliable data entry into your software as if from your keyboard. Internal unit for PC, XT, AT, PS/2-N30, RS-232 unit for DOS and non-DOS systems (including all PS/2). Stainless steel wands and laser interfaces. Powerful bar code and text printing software with formatting flexibility.
Seagull Scientific Systems
601 University Ave., Suite 150
Sacramento, CA 95825
(916) 386-1776

BUSINESS OPPORTUNITIES

CASH FOR YOUR PROGRAMS

No gimmicks. No fees. We don't want YOUR money. We'll pay YOU! We're looking for a few ORIGINAL, well-structured PC/MS-DOS programs to buy. All categories except educational and games. Here is your golden opportunity to cash in on your ideas. Write for submission agreement. Quick response.
CASH BUYER
SOFTWARE PROCUREMENT DIVISION
P.O. Box 91073
San Diego, CA 92109

COMPUTER INSURANCE

SAFEWARE

Insures your computer. Safeware provides full replacement of hardware, media, and purchased software. As little as $39 a year provides comprehensive coverage. With our blanket coverage, no lists of equipment are needed. One phone call does it all! Call 8am - 10pm ET Monday thru Friday. (Sat. 9 - 5)
SAFEWARE, The Insurance Agency Inc.
2929 N. High Street, P.O. Box 02211
Columbus, Ohio 43202

800-848-3469 National
614-262-0559 Ohio

CIRCLE 384 ON READER SERVICE CARD

DATA CONVERSION

FILE CONVERSION EXPERTS

TSC

Text Sciences Corporation
5430 San Fernando Road
Glendale, CA 91203
(818) 247-0792

We will convert your files to any computer (tapes, disks, cards) faster and easier than anyone! Try us. Since 1979.

CIRCLE 385 ON READER SERVICE CARD

DATA CONVERSION

TAPE/DISK CONVERSIONS

Conversion services to or from over 1000 computer systems:

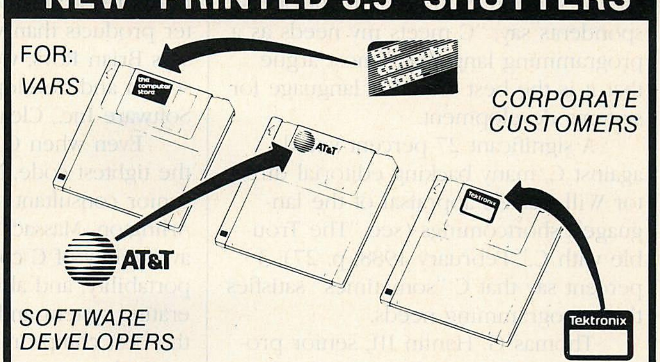
- Magtapes
- Micro Computers
- Mini Computers
- Word Processors
- Typesetters

Our conversion capabilities surpass most in the industry.
PIVAR COMPUTING SERVICES, INC.
165 Arlington Hgts. Rd. #T
Buffalo Grove, IL 60089
(312) 459-6010

PC TECH JOURNAL MARKETPLACE is a special economical section for product and service listings.
Listings are grouped by category and sold by column inches.
Second color option available.
Standard Directory Listings are also available for a minimum of 3 issues at $220 per issue ($660 total).

SOFTWARE DUPLICATION

"NEW" PRINTED 3.5" SHUTTERS



Permanent Identification of your Company, Information, or Products.

SOFTWARE DUPLICATION/PACKAGING

3.5" / 5.25" / HIGH DENSITY

WEST COAST TELECOM
CALL: (503) 620-1888

CIRCLE 386 ON READER SERVICE CARD

TECH MARKETPLACE . . .
Second Color Option Available

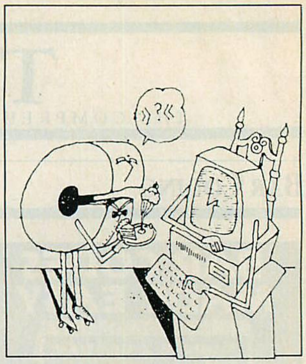
Tech Marketplace . . . the comprehensive guide to products and services for the MS DOS market.

Call (212) 503-5115

JORDENE ZEIMETZ

PROFESSIONAL VIEWPOINT

The C controversy: two of three survey respondents endorse C, despite hard knocks against it.



No programming language is perfect. The much maligned, yet popular C is a case in point. C is repeatedly criticized for being complicated, time consuming, error prone, and difficult and expensive to maintain—all in all, an unproductive and unprofitable programming tool.

Yet, in an informal survey of *PC Tech Journal* readers, C amassed widespread support. Seventy percent of respondents say, "C meets my needs as a programming language;" most argue that it is the best available language for systems development.

A significant 27 percent voted against C, many backing editorial director Will Fastie's appraisal of the language's shortcomings (see "The Trouble with C," February 1988, p. 27); 3 percent say that C "sometimes" satisfies their programming needs.

Thomas G. Hanlin III, senior programmer at Hammerly Computer Services Inc., Laurel, Maryland, echoes the sentiments of the opposing 27 percent: "C is hard to debug. It lacks many important features (although newer C versions for the PC have been improving drastically). C is lousy at handling strings, and is mediocre at displaying information. It has no communications capabilities or error trapping."

Others, such as Stuart Jones M.D., chief of nuclear medicine at Lehigh Valley Hospital Center, Allentown, Pennsylvania, are looking to future alternatives. Jones says that C "has a combination of the worst features of assembly language and high-level languages. What's needed is a new language with clear syntax, low-level capabilities, a good MS-DOS compiler, and an ANSI standard from the start."

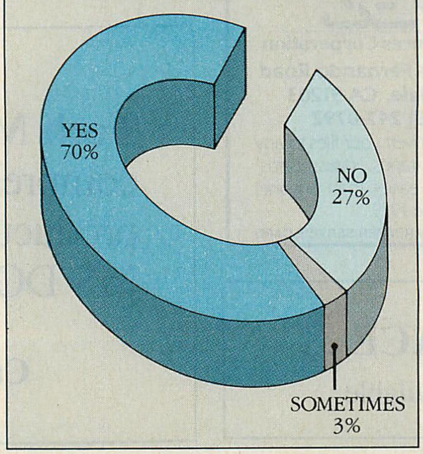
Advocates rank portability, speed of coding *and* execution, flexibility, and power as C's top assets. Following in importance are C's efficiency, modularity, concise and clean syntax, good access to machine functions, balance between structure and creativity, and a wealth of good compilers, editors, debuggers, and library support.

Supporters say they use C for all or most of their business, data management, graphics, accounting, financial, and scientific applications. "C has the right combination of speed, freedoms, restrictions, power, efficiency, consistency, and flexibility that allows me to be more productive and produce better products than with most languages," says Brian Roys, vice president of research and development at Practical Software Inc., Clearwater, Florida.

Even when C does not generate the tightest code, Michael Benveniste, senior consultant at Dossier Resources, Arlington, Massachusetts, says, "The availability of C compilers and libraries, portability, and ability to reach the operating system and hardware provides the best compromise when working on more than one machine."

Another C advocate, Gerry Danen, president of Danen Computer Consulting, Edmonton, Alberta, Canada, disputes the criticism that programming in C is difficult: "With proper use of a function library, an ease of programming can be achieved that anti-C folks claim C does not have."

Does C meet your needs as a programming language?



C's potential for enhancing developer creativity draws extensive support. "C allows competent programmers to play fast and loose and, by linking in assembly routines, to exploit the odds and ends of various operating systems and hardware configurations," says James Thomas, president of Enhanced Systems, Springfield, Louisiana.

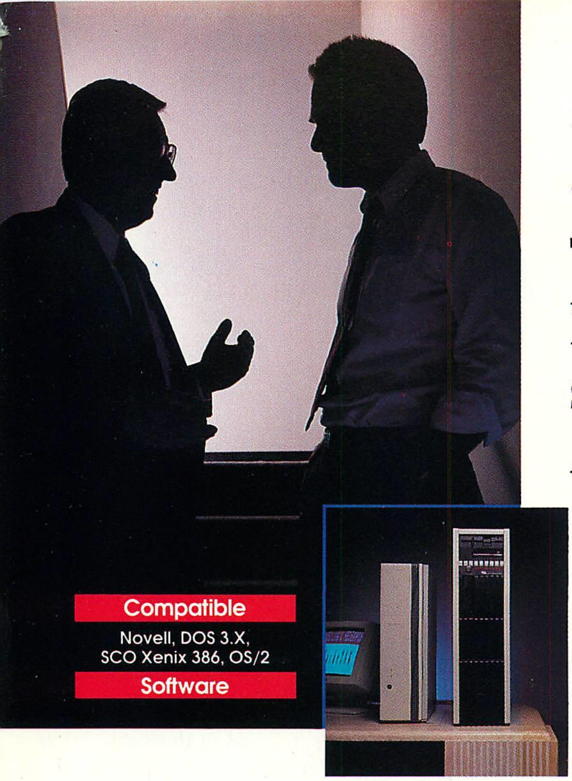
Phil Albert, president of Galt Engineering Inc., Pasadena, California, adds, "With some languages, when you think of a design and try to express that design in code, you will find that it is syntactically impossible to do. With C, you write code that comments itself. All complicated structures are due to design, not work-arounds to get programs to compile."

Of the charge that C is error prone, Karl Peterson, engineering manager at T.E.N. Inc., Provo, Utah, says: "I know how to program, and with C I don't spend hours chasing syntax errors that have nothing to do with the logic of the program but are required by the compiler."

Steven Jones, president of Micro Decisions, Fort Worth, Texas, says, "C is difficult to learn, but I would expect something with this power and flexibility would be."

Once competency is achieved, concludes Mike Wengler, special projects manager at Computer Related Services, Virginia Beach, Virginia, "C programmers are as productive as those writing programs in other languages."

Beyond expected differences in personal opinion, this striking discord on the merits and shortcomings of C can be explained in two ways. First, C requires considerable study and discipline—effort that many critics feel detracts from productivity. Second, C supporters are focusing on current usage and experience, while C critics are looking to future alternatives.



POWER

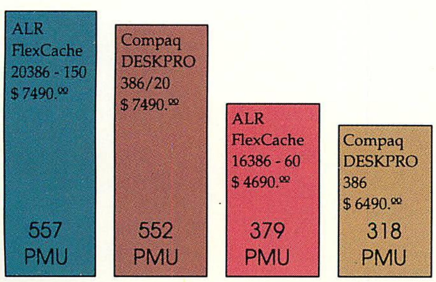
The FlexCache 386 series from ALR

"Spending more money doesn't guarantee that you'll end up with the system you need. Make some serious comparisons of features and prices and then get the most for your money."

At Advanced Logic Research (ALR) the philosophy of getting more for your money has become an unbroken tradition. A tradition that has been recognized by all the major trade journals with excellent reviews. Based on the ALR 386/220, PC magazine's choice as "The Best of 1987", ALR extends its product line with the FlexCache 386 series. Now the fastest PCs available, the FlexCache 386 series approach minicomputer proportions and offer new ways to get the most for your money.

ALR's advanced FlexCache architecture provides a wide open, high-speed data channel for up to 60% faster CPU/memory data through-put than the IBM PS/2 model 80-071 with the much touted microchannel architecture.

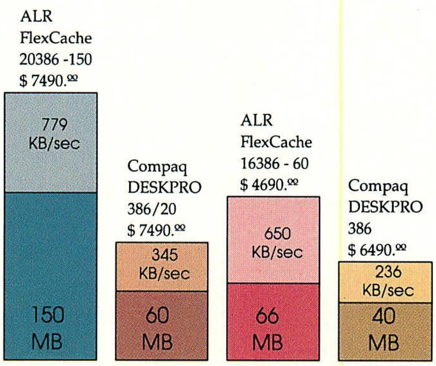
The cache memory controller can eliminate wait-states up to 95% of the time. This powerful blend of enhancements allows a FlexCache 16MHz CPU to move data along as fast as many 20MHz CPUs and a FlexCache 20MHz CPU to move data even faster than a Compaq DESKPRO 386/20™.



Power Meter Performance Index

The FlexCache 386 series comes equiped with the most fixed disk capacity for your money. The FlexCache 20386 will give you an extra 45,000 pages of document disk storage for free when you compare it to the performance and price of Compaq's DESKPRO 386/20 model 60.

Full track data transfering (1:1 interleave) plus ESDI (Enhanced Small Device Interface) look-ahead buffering, turns what used to be a data traffic bottleneck into a super high-speed corridor. FlexCache 386 systems achieve transfer rates twice that of Compaq's DESKPRO 386 and 386/20 model 60.

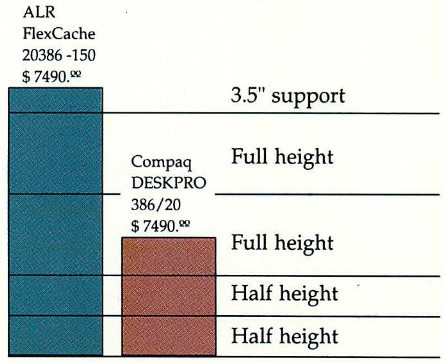


Fixed Disk Capacities & Transfer Rates

The FlexCache series offers power and expansion possibilities not easily exhausted. The minicomputer-style chassis of FlexCache 20386 offers space for five internal peripheral devices, allowing more data storage devices than any other PC available.

Circle No. 116 for Re-seller, No. 118 for End-user.

With the money you save on a FlexCache system you can afford additional data storage options.



Internal Device Support

To make some serious feature and cost comparisons give ALR a call at (800) 366-2574 or (714) 581-6770 for the name of the dealer nearest you.

FlexCache 386 Series Specifications

- ALR designed and proven multi-layer system board
- Socketed for 80387 support
- 1MB 32 bit RAM, expandable to 2MB on system board
- 0-wait-state cache memory controller with its own 32KB of high-speed (35 ns) static RAM
- Enhanced 101 keyboard
- Phoenix BIOS
- Dual drive support
- OS/2 compatible

FlexCache 16386 Model 60 ...$4690⁰⁰

- FlexCache 386 series specifications
- 80386 CPU with 16MHz system clock
- 80387 support with 16MHz clock
- 66Mbyte <30ms hard disk

FlexCache 16386 Model 100 ...$5690⁰⁰

- FlexCache 386 series specifications
- 80386 CPU with 16MHz system clock
- 80387 support with 16MHz clock
- 100Mbyte <30ms hard disk

FlexCache 20386 Model 100 ...$6490⁰⁰

- FlexCache 386 series specifications
- 80386 CPU with 20MHz system clock
- 80387 support with 20MHz clock
- 100Mbyte <30ms hard disk

FlexCache 20386 Model 150 ...$7490⁰⁰

- FlexCache 386 series specifications
- 80386 CPU with 20MHz system clock
- 80387 support with 20MHz clock
- 150Mbyte <23ms, track buffered, ESDI hard disk

FlexCache 20386 Model 300 ...$9990⁰⁰

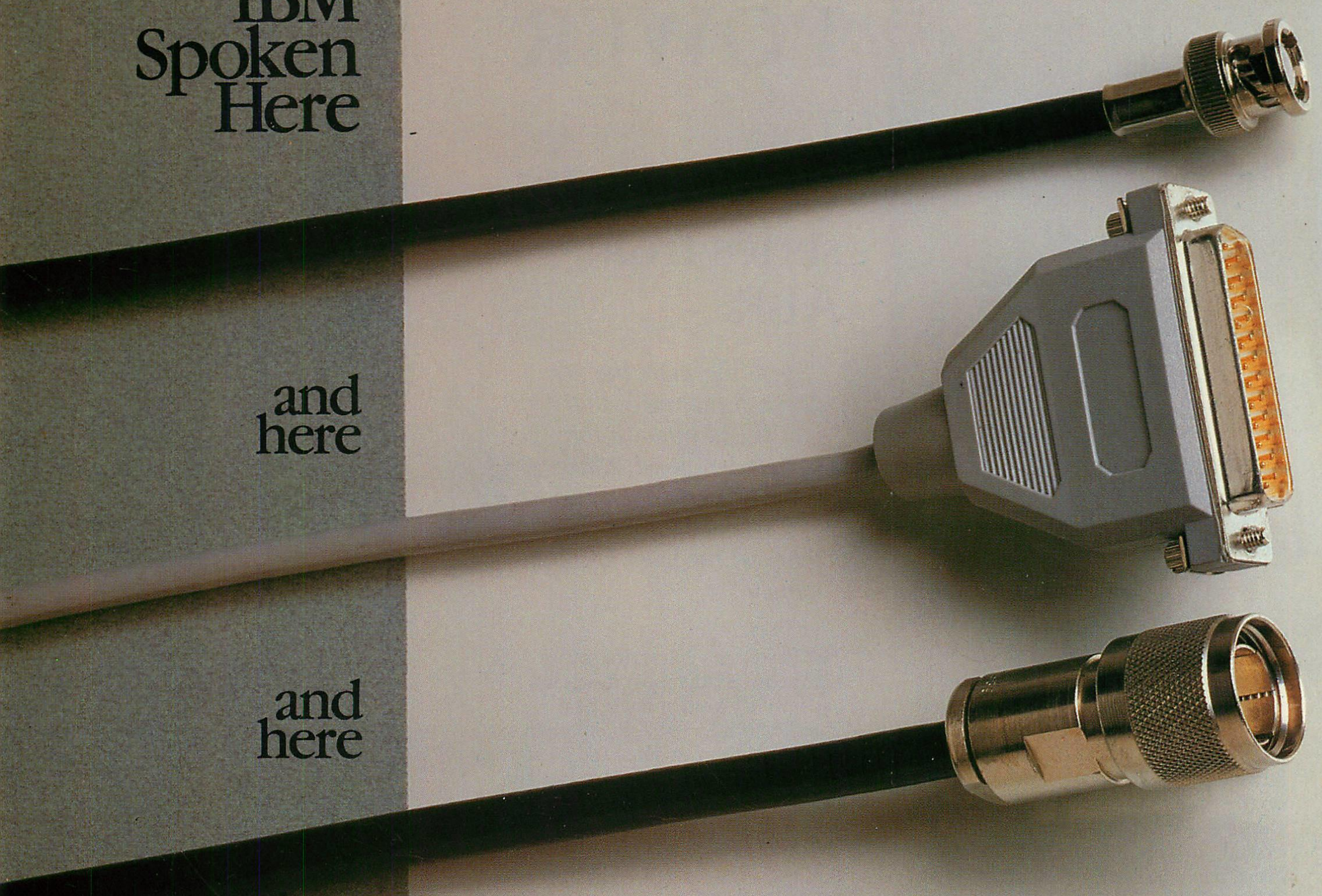
- FlexCache 386 series specifications
- 80386 CPU with 20MHz system clock
- 80387 support with 20MHz clock
- 300Mbyte <20ms, track buffered, ESDI hard disk

Advanced Logic Research, Inc.

10 Chrysler, Irvine, CA 92718
714-581-6770
FAX: 714-581-9240
Telex: 5106014525,
Answer back Advanced Logic

From Asia or Europe call:
ALR/Wearnes Technology
(65) 2592521 (Singapore)
Telex: RS38113 WRNTEC

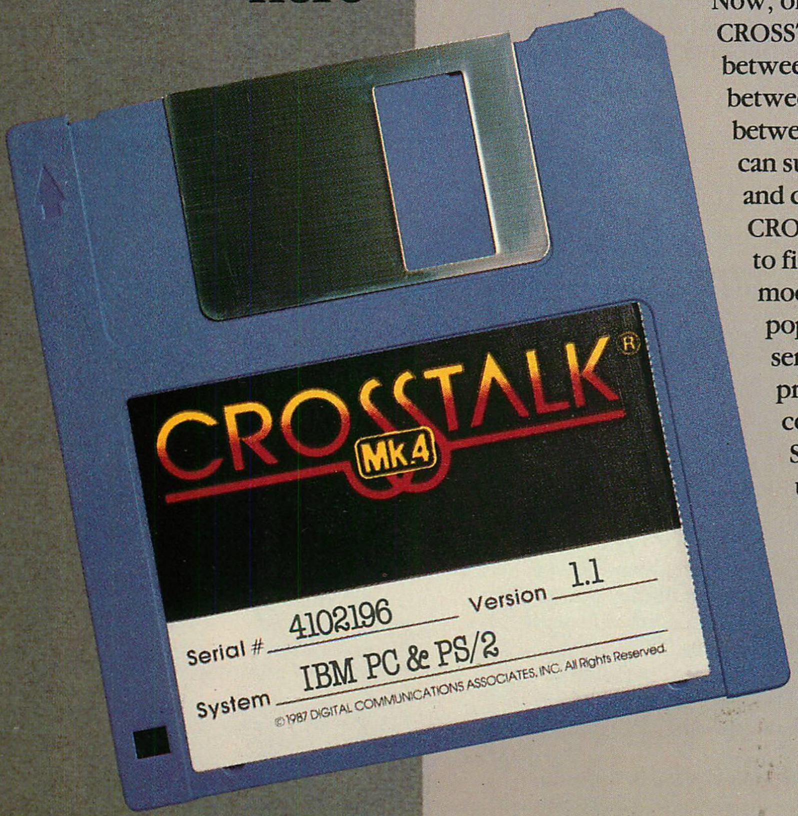
IBM Spoken Here



and here

and here

and here



Whatever dialect of IBM you need to speak, CROSSTALK® Mk. 4 makes the connection.

Now, one program does the job that used to require several. CROSSTALK® Mk. 4 allows high-speed direct communications between PCs and minicomputers, or (with an IRMA™ board) between your PC and an IBM Mainframe, or (with Smart Alec™) between your PC and IBM System 3x's. If you like, CROSSTALK can support all of these sessions (and others) simultaneously, and display each session in its own window.

CROSSTALK Mk. 4 emulates all the terminals you're likely to find useful. That includes IBM 3101 (page and character modes), IBM 525x, IBM 529x, IBM 327x, as well as many popular async terminals like the DEC VT100 and VT220 series. CROSSTALK Mk. 4 includes the powerful CASL™ programming language, which allows you to automate communications applications quickly and easily.

So if you're used to thinking of CROSSTALK just to use with a modem, you're missing some important connections. Ask your dealer for details, or write:

dca® Digital Communications Associates, Inc.
1000 Holcomb Woods Parkway / Roswell, Georgia 30076
1-800-241-6393

CROSSTALK® COMMUNICATIONS

CIRCLE NO. 167 ON READER SERVICE CARD